TPC Benchmark® C
Full Disclosure Report

HP Integrity Superdome

using Microsoft SQL Server 2005 Enterprise Edition 64-bit
Service Pack 1
on Microsoft Windows Server 2003, Datacenter Edition (64-bit)
SP1

Second Edition
November 29, 2005

Second Edition - November 29, 2005

Hewlett-Packard Company believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Hewlett-Packard Company assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Hewlett-Packard Company provides no warranty of the pricing information in this document.

Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC Benchmark® C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Hewlett-Packard Company does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC ®) or normalized price/performance ($/tpmC ®). No warranty of system performance or price/performance is expressed or implied in this report.

© Copyright Hewlett-Packard Company 2005.

All rights reserved. Permission is hereby granted to reproduce this document in whole or in part provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

Printed in U.S.A., November 29, 2005

HP and HP StorageWorks are registered trademarks of Hewlett-Packard Company.

Microsoft Windows NT, SQL Server and COM+ are registered trademarks of Microsoft Corporation.

Intel, Pentium, Xeon and Itanium 2 are registered trademarks of the Intel Corporation.

TPC Benchmark, TPC-C, and tpmC are registered certification marks of the Transaction Processing Performance Council.

All other brand or product names mentioned herein are trademarks or registered trademarks of their respective owners.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark® C test conducted on the HP Integrity Superdome in a client/server configuration, using Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 and Microsoft COM+ Transaction Monitor. The operating system used for the benchmark was Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1.

TPC Benchmark® C Metrics

The standard TPC Benchmark ® C metrics, tpmC® (transactions per minute), price per tpmC ® (three year capital cost per measured tpmC®), and the availability date are reported as required by the benchmark specification.

Standard and Executive Summary Statements

The following pages contain the executive summary of the benchmark results for the HP Integrity Superdome system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP Integrity Superdome	Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1	Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$5,931,555 USD	1,231,433 tpmC	$4.82 USD per tpmC	May 5, 2006

Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per tpmC® , were audited by Lorna Livingtree of Performance Metrics to verify compliance with the relevant TPC specifications.

Hewlett Packard Company	HP Integrity Superdome Enterprise Server		TPCC Rev 5.5 October 2005 Report Date November 29 2005			
Description	Price Key	Part Numbr	Unit Price	0	Extended Price	3 Yr Maint Price
Superdome left cabinet	1	A5201A, Opt. 429	$236,716	1	$236,716	
Superdome right cabinet	1	A5202A, Opt. 429	$251,200	1	$251,200	
IPF Superdome Cell Board (sx1000)	1	A6866A	$19,200	16	$307,200	
3 Year Svc & Support Price (Hardware and Software)						$888,659
HP Superdome Itanium2 1.6 GHz 9M 2 CPU	1	AD003A	$41,800	32	$1,337,600	
Superdome 128GB Memory Pack	1	AB547A	$282,880	8	$2,263,040	
PCI-x I/O chassis	1	A6864A	$16,805	8	$134,440	
Core I/O card	1	A6865A	$1,045	1	$1,045	
Gig E - TX Adapter	1	A7061A	$270	1	$270	
Graphics USB Card	1	A6869A	$299	1	$299	
FC-HBA 2GB, 1 Channel (LP1050)	1	AB467A	$1,520	4	$6,080	
Qlogic QLA-2350 Fibre-Channel VI Adapter	1	A390975	$1,256	16	$20,096	
HP Smart Array Controller 6402 (boot)	1	A9890A	$1,669	1	$1,669	
HP Smart Array Controller 6404 (incl boot)	1	A9891A	$2,549	30	$76,470	
36GB, 15krpm Ultra3 Wide disk (Incl Boot)	1	286776-B22	$299	1686	$504,114	
36GB, 15krpm Ultra3 Wide disk (10% Spares)	1	286776-B22	$299	169	$50,531	
Storageworks MSA30 SB (incl boot)	1	302969-B21	$2,829	121	$342,309	
Storageworks MSA30 SB (10% Spares)	1	302969-B21	$2,829	13	$36,777	
HP9000 Standard Rack System E41	1	A4902A	$1,910	14	$26,740	
HP Power Distribution Unit 120-240V	1	A5137AZ	$145	27	$3,915	
Modular Storage Array 1000	1	201723-B22	$6,995	4	$27,980	
MSA1000 controller	1	218231-B22	$4,290	4	$17,160	
Storage MSA1000 Support 3yrs 24x7 4hr incl disks	1	HA110A3 #6FG	$3,222	4		$12,888
146GB, 10krpm disk	1	286716-B22	$499	56	$27,944	
5M LC to LC Cable Kit	1	221692-B22	$82	4	$328	
TA5300 Enclosure for DAT tape	1	C7508AZ	$729	1	$729	
DVD Rom drive	1	Q1592A	$649	1	$649	
DAT Tape	1	C7497B	$1,049	1	$1,049	
sx1000 Superdome SMS, rack	1	A9802B	$6,500	1	$6,500	
	Server Subtotal				**$5,682,850**	**$901,547**
Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1	1	T2372A opt064	$128,000	1	$128,000	
Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pa	2	810-03134	$23,432	64	$1,499,648	$245
	Server Software Subtotal				**$1,627,648**	**$245**
DL140 Dual 3.06GHz Xeon 1GB 80GB ATA	1	356533-B21	$1,799	31	$55,769	
HP 3 Yr. 4h 24x7 ProLiant DL140 Hardware Support	1	U9510E	$419	12		$5,028
1GB PC2100 ECC DDR	1	351109-B21	$419	31	$12,989	
HP Mouse	1	P5304M	$28	1	$28	
HP Enhanced Keyboard (USB/PS2)	1	DC852A#ABA	$29	1	$29	
S7540 17in CRT Monitor	1	PF997AA	$149	1	$149	
Startech 32port KVM switch	4	5150	$1,599	1	$1,599	
KVM Cable	4	5250	$9	31	$279	
Qlogic QLA-2350 FC VI Adapter (incl 10% spares)	1	A390975	$1,256	35	$43,960	
5M LC to LC Cable Kit	1	221692-B22	$82	48	$3,936	
	Client Subtotal				**$118,738**	**$5,028**
Microsoft Windows 2003 Server	2	P73-00295	$719	31	$22,289	
Microsoft Visual C++ Standard	2	254-00170	$109	1	$109	
Microsoft Problem Resolution	2		$245	1		$245
	Client Software Subtotal				**$22,398**	**$ 245**
Qlogic SANBox 5200 16 2GB Port (incl spares)	3	SB5200-16A	$4,371	3	$13,113	
SANbox 5200 Prime 7x24x4 Maint. Upgrade, 1-Year	3	Prime-SB5200	$1,327	9		$11,943
Finisar 2GB SFP (incl 10% spares)	3	FTRJ8519P1BNL	$84	52	$4,368	
	Connectivity Subtotal				**$17,481**	**$11,943**
HP's Large Configuration Discount *	*** Discounts:**				**-$2,098,024**	**-$358,544**
	Total:				**$5,371,091**	**$560,464**
Price Key: 1-HP, 2-Microsoft, 3-SANMarketplace.com 4 - KVMs.com	**3 year cost of ownership:**					**$5,931,555**
			tpmC:			1,231,433
* A 35.94% discount was based on the overall value of the specific components from HP (Price Key 1) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration			**$/tpmC:**			**$4.82**

Numerical Quantities Summary for HP Integrity Superdome

MQTH, Computed Maximum Qualified Throughput **1231433 tpmC**

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.92s	13.29s	0.40s
Payment	0.89s	12.23s	0.37s
Order-Status	0.95s	11.79s	0.42s
Delivery (interactive portion)	0.42s	10.65s	0.20s
Delivery (deferred portion)	0.19s	10.72s	0.14s
Stock-Level	1.07s	12.04s	0.52s
Menu	0.42s	10.65s	0.20s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

New-Order	44.97%
Payment	43.01%
Delivery	4.01%
Stock-Level	4.01%
Order-Status	4.01%

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18.00s	18.01s	20.78s	0.00s	12.07s	121.28s
Payment	3.00s	3.02s	5.28s	0.00s	12.06s	120.91s
Order-Status	3.00s	3.02s	5.23s	0.00s	10.06s	100.53s
Delivery (interactive)	3.00s	3.02s	5.17s	0.00s	5.07s	50.53s
Stock-Level	3.00s	3.02s	5.27s	0.00s	5.06s	50.53s

Test Duration

Ramp up time	110 minutes
Measurement interval	120 minutes
Transactions during measurement interval	341819601
Ramp down time	5 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	28 minutes

Table of Contents

Abstract....1
Overview....1
TPC Benchmark® C Metrics....1
Standard and Executive Summary Statements....1
Auditor....1
Table of Contents....4
Preface....6
Document Structure....6
TPC Benchmark® C Overview....6
System Overview....7
General Items....8
Test Sponsor....8
Application Code and Definition Statements....8
Parameter Settings....8
Configuration Diagrams....8
Chapter 1 Logical Database Design....10
1.1 Table Definitions....10
1.2 Physical Organization of the Database....10
1.3 Insert and Delete Operations....10
1.4 Partitioning....10
1.5 Replication, Duplication or Additions....10
Chapter 2 Transaction and Terminal Profiles....11
2.1 Random Number Generation....11
2.2 Input/Output Screen Layout....11
2.3 Priced Terminal Feature Verification....11
2.4 Transaction Statistics....11
2.5 Presentation Manager or Intelligent Terminal....12
2.6 Queuing Mechanism....12
Chapter 3 Transaction and System Properties....13
3.1 Transaction System Properties (ACID Tests)....13
3.2 Atomicity Tests....13
3.2.1 COMMIT Transaction....13
3.2.2 ROLLBACK Transaction....13
3.3 Consistency Tests....13
3.4 Isolation Tests....14
3.5 Durability Tests....14
3.5.1 Loss of Data / Loss of Log....14
3.5.2 Loss of System / Memory....15
Chapter 4 Scaling and Database Population....16
4.1 Database Layout....16
4.2 Initial Cardinality of Tables....22
4.3 60 Day Space....22
4.3.1 Transaction Log Space Requirements....22
4.4 Type of Database Used....23
4.5 Database Mapping....23
Chapter 5 Performance Metrics and Response Time....24
5.1 Throughput....24
5.2 Response Times....24
5.3 Keying and Think Times....24
5.4 Response Time Frequency....25
5.4.1 New Order Response Time....25
5.4.2 Payment Response Time Distribution....26

5.4.3 Order Status Response Time 27
5.4.4 Delivery Response Time Distribution 28
5.4.5 Stock Level Response Time 29
5.4.6 Response Time Versus Throughput 30
5.4.7 New Order Think Time Distribution 31
5.4.8 Throughput Versus Time Distribution 32
5.5 Steady State Determination 32
5.6 Work Performed During Steady State 32
5.6.1 Checkpoint 33
5.6.2 Checkpoint Conditions 33
5.6.3 Checkpoint Implementation 33
5.7 Measurement Period Duration 33
5.8 Regulation of Transaction Mix 33
5.9 Transaction Mix 33
5.10 Transaction Statistics 34
5.11 Checkpoint Count and Location 34
Chapter 6 SUT, Driver and Communications Definition 35
6.1 RTE Description 35
6.2 Emulated Components 35
6.3 Functional Diagram 35
6.4 Networks 35
6.5 Operator Intervention 35
Chapter 7 Pricing 36
7.1 System Pricing 36
7.2 General Availability, Throughput and Price Performance 36
7.3 Country Specific Pricing 36
7.4 Usage Pricing 36
Chapter 8 Audit 37
8.1 Auditor's Information 37
Appendix A Source Code 41
Appendix B Database Load 185
B.1 Database Options 211
B.2 Table definitions 212
B.3 Stored Procedures 217
Appendix C Tunable Parameters 373
C.1 Microsoft SQL Server 8.0 Configuration Parameters 459
C.2 Client System Configuration Parameters 460
C.3 RTE Input Parameters 482
Appendix D 60 Day Space Requirements 502
Appendix E 3rd Party Pricing 503

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP Integrity Superdome using Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1. It meets the requirements of the TPC Benchmark ® C Standard Specification, Revision 5.5 dated October 2005. TPC Benchmark® C was developed by the **T**ransaction Processing **P**erformance **C**ouncil (TPC). It is the intent of this group to develop a suite of benchmarks to measure the performance of computer systems executing a wide range of applications. Hewlett-Packard Company and Microsoft, Inc. are active participants in the TPC.

TPC Benchmark® C Overview

TPC Benchmark ® C is an **O**n **L**ine **T**ransaction **P**rocessing (OLTP) workload. It is a mixture of read-only and update intensive transactions that mimic the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention of data access and update

The performance metric reported by TPC-C ® is a "business throughput" measurement of the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C ® (tpmC®). To be compliant with the TPC-C ® standard, all references to tpmC ® results must include the tpmC ® rate, the associated price-per-tpmC®, and the availability date of the priced configuration.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C® approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments are not recommended.

Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

Hewlett-Packard Company does not warrant or represent that a user can or will achieve performance similar to the benchmark results contained in this report. No warranty of system performance or price/performance is expressed or implied by this report.

System Overview

The hardware configuration used in this TPC-C test was based on the HP Integrity Superdome. The full configuration was built by adding additional memory, additional disk adapters and drives. The operating system used on the server was Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1 and the database was Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1.

The processor architecture of the HP Integrity Superdome was designed for the Intel Itanium 2 9M processor. The HP Integrity Superdome used in this test was powered by 64 1.6GHz Intel Itanium 2 9M processors, each with 9 MB of 3rd level cache.

This configuration used 1024 GB of HP SDRAM. This was achieved by using 512 2GB DIMMs.

The operating system, all executables and libraries, the master database, and swap space were contained in one 18GB hard disk pair (mirrored RAID 1), attached to an HP SmartArray 6402 controller. 1 additional mirrored pair of 36GB drives were used for utility storage of scripts, the build environment, etc. (An additional pair of 18GB disk was present in the system and priced, but just contained alternate boot configurations and was not used during the benchmarks).

The database log drive storage was located on 4 HP MSA1000's. Each MSA1000 held 14 146MB Ultra320 SCSI hard drives. The MSA1000 disk arrays were connected to the Integrity Superdome using 4 Emulex 1050 Fibrechannel HBA's. The disks were configured as RAID 1+0, and two battery backed up disk array controller caches were enabled and mirrored on each MSA1000 (90% write, 10% read). The 4 Arrays were then software striped using Windows Dynamic Disk Management.

The TPC-C database storage consisted of 1064 HP 36GB 15KRPM disk drives and 616 HP 18GB 15KRPM disk drive (measured configuration). 120 HP StorageWorks MSA30 disk arrays were used to connect the disks. Each MSA30 holds 14 disks. The MSA30 enclosures were attached in groups of 4 to each SmartArray 6404 Controller using standard SCSI cabling. The 6404 uses the 4 enclosures as two groups of two, or 28 discs in each. The discs were configured as a RAID0 Array spanning all 28 discs in each array, for a total of 60 RAID0 data base arrays. The SmartArray caches were disabled on the RAID0 partitions. Additionally, a RAID5 volume was configured on each set of 28 discs as fault tolerant backup for the data base. Windows partitions were created on the RAID0 volumes to contain the CS and MISC SQL filegroups. The partition sizes were the same on all 60 volumes, regardless of whether the underlying arrays were 18gb or 36 gb disks.

Performance was monitored comparing the 18GB and 36GB drives and proved to be compliant with the rules for substitution. The 36GB drives were priced.

Each of the 32 clients is an HP Proliant DL140 with two Intel Xeon at 3.06 GHz, 2048 MB RAM and one 80.1GB ATA hard disk, running Microsoft Windows 2003 Server with IIS 6.

The server and web-clients were networked together using QLogic VIA/SAN. The network bandwith between the web-clients and the database server was 2GBps. 31 remote terminal emulators (RTEs) emulated 993,500 users executing the standard TPC-C workload. Each web-client had two embedded Gigabit LAN adapters, one of which was was used to connect to the RTEs running in Gigabit mode. HP DL140's were also used as the RTE emulators.

Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 was configured to utilize "soft NUMA", a feature that allows network connections to be affined to specific groups of CPUs (this is independent of the hardware NUMA features of the HP Integrity Superdome). SQL Server was configured with 33 SoftNuma nodes. 31 of these nodes were configured with 2 CPUs each. The 32nd SoftNuma node as configured with one CPU and used for the checkpoint process. The 33rd SoftNuma node, also with one processor, was reserved for the Log Writer thread. A script doing continuous checkpoints of 1700 seconds (28 minutes, 20 seconds) was started on one of the server after steady state was reached with a connection port that connected to the 33rd Numa Node. This allowed the main checkpoint process to run on that single processor, which directed the processing of the checkpoint tasks that SQL assigned to each of the other 31 SoftNuma Nodes.

General Items

Test Sponsor

A statement identifying the sponsor of the Benchmark and any other companies who have participated.

Hewlett-Packard Company was the test sponsor of this TPC Benchmark C.

Application Code and Definition Statements

The application program must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input/output functions.

The Section 3.0 entitled Clause 3 Related Items contains a brief discussion of the database design and loading. The database definition statements, distribution across disk drives, loading scripts, and tables are provided in Appendix B.

The program that implements the TPC Benchmark C translation and collects appropriate transaction statistics is referred to as the Remote Terminal Emulator (RTE) or Driver program. We have used the Microsoft BenchCraft RTE program that emulated a set of users entering TPC-C transactions through web browsers, and communicating with web-client machines running the Microsoft Internet Information Server (IIS) web server. The web-client machines used the COM+ transaction monitor (TM) to communicate with the database server.

On each web-client machine, IIS loads a custom Microsoft Internet Information Server Application Programming Interface dynamic link library (ISAPI DLL) application program that communicates with the emulated web browsers through the HTTP protocol and the database server through the COM+ TM and the Microsoft ODBC interface. The application supplies fill-in screens to the user for each transaction, then parses the data in each request, and makes a call on SQL Server through the COM+ layer, which manages a set of ODBC connections to the database server. The resulting data is passed back to the application where it is formatted into HTML and sent back to the user's browser. The *delivery* transaction is handled directly from the application to the database without the use of COM+.

Parameter Settings

Settings must be provided for all customer-tunable parameters and options which have been changed from the default found in actual products; including but not limited to:

- *Database options*
- *Recover/commit options*
- *Consistency/locking options*
- *System parameter, application parameters, and configuration parameters.*

Appendix C contains all the database and operating system parameters used in this benchmark in addition to all the hardware configuration details.

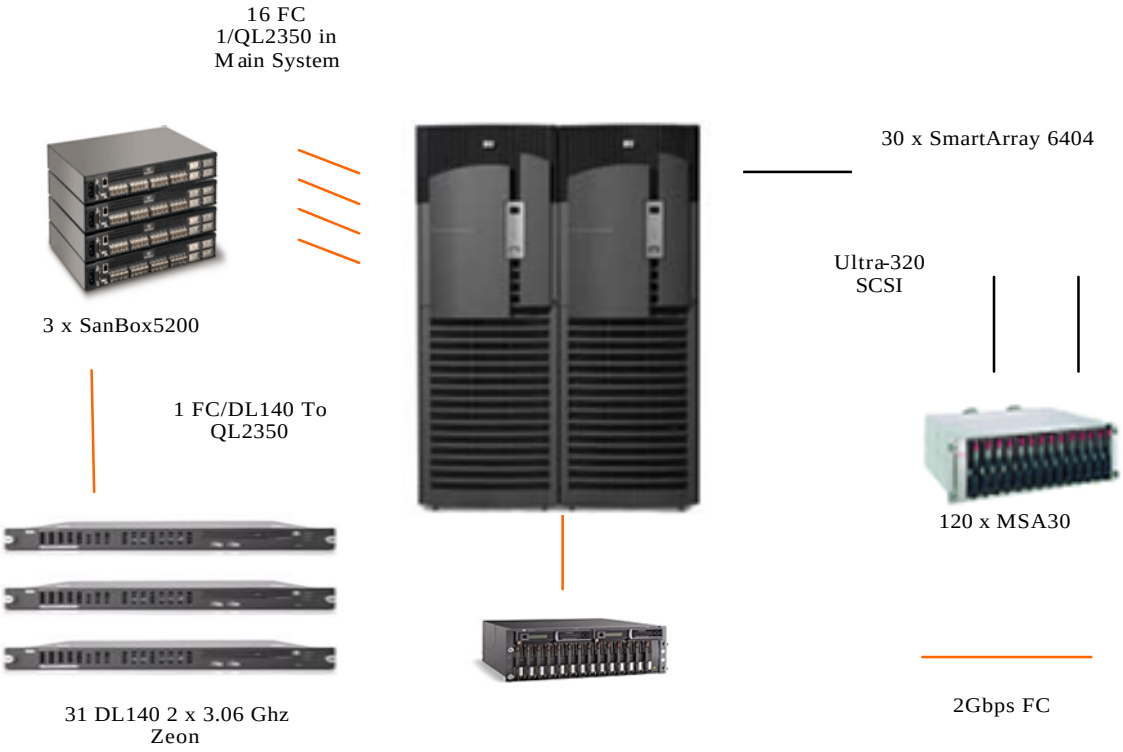
Appendix D contains the 60 day space calculations.

Configuration Diagrams

Diagrams of both the measured and priced system must be provided, accompanied by a description of the differences.

The measured and priced client/server configuration is shown in Figures 1.

Figure 1. Measured and Priced Configuration



The measured configuration used a mix of 36GB 15KRPM and discontinued 18GB 15KRPM SCSI as noted in the system description, above. The 36GB 15KRPM disks were priced.

Chapter 1 Logical Database Design

1.1 Table Definitions

A listing must be provided for all table definition statements and all other statements used to set up the database.

Appendix B contains the code used to define and load the database tables.

1.2 Physical Organization of the Database

The physical organization of tables and indices within the database must be disclosed.

The measured database configuration used a total of 1738 disks, which included 1064 HP 36GB 15KRPM and 616 HP 18GB 15KRPM drives for data, 56 HP 146GB, 10krpm U320 drives for log, and two 36GB drives (mirrored) for the operating system.

Part of the space on each of the 60 database disk arrays was configured as 1 RAID0 volume over 28 18GB or 36GB drives. Each volume held 2 partitions, one for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored. The remainder of the disc space on each of the 60 Arrays was configured as a RAID5 volume over all 28 18GB or 36GB drives. Each volume had 2 partitions, containaining backups of the data base.

1.3 Insert and Delete Operations

It must be ascertained that insert and delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

All insert and delete functions were fully operational and verified during the entire benchmark.

1.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C Benchmark, any such partitioning must be disclosed.

Partitioning was not used on any table.

1.5 Replication, Duplication or Additions

Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used.

Chapter 2 Transaction and Terminal Profiles

2.1 Random Number Generation

The method of verification for the random number generation must be disclosed.

The random number generation was done internal to the Microsoft BenchCraft RTE program, which was audited independently.

2.2 Input/Output Screen Layout

The actual layout of the terminal input/output screens must be disclosed.

The screen layouts are based on those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C® Standard Specification.

2.3 Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal features were verified by allowing the auditor to manually execute each of the five transaction types, using the Microsoft Internet Explorer.

2.4 Transaction Statistics

The transaction profiles must be disclosed as per Clauses 8.1.3.5 through 8.1.3.10.

Table 1 shows the transaction statistics.

Table 1. Transaction Statistics

Type	Item	Value
New Order	Home warehouse items	99.00%
	Remote warehouse items	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	85.00%
	Remote warehouse	15.00%
	Non primary key access	60.01%
Order Status	Non primary key access	60.06%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.97%
	Payment	43.01%
	Delivery	4.01%
	Stock Level	4.01%
	Order Status	4.01%

2.5 Presentation Manager or Intelligent Terminal

Any usage of presentation managers or intelligent terminals must be explained.

Comment 1*: The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations, message bundling, and local storage of TPC-C rows.*
Comment 2*: This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

Application code running on the web-client implemented the TPC-C® user interface. Screen manipulation commands in the form of HTML were downloaded to the web browser, which handled input and output presentation graphics. A listing of this code is included in Appendix A. Microsoft Internet Information Service assisted in the processing and presentation of this data.

2.6 Queuing Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed.

The application creates a semaphore-based thread pool consisting of a user-specified number of threads, which open ODBC connections on the database. When a *delivery* transaction is posted, one of these threads makes the database call while the transaction's original thread returns control to the user. Upon completion, the delivery thread writes an entry in the delivery log and returns to the thread pool.

The source code is listed in Appendix A.

Chapter 3 Transaction and System Properties

3.1 Transaction System Properties (ACID Tests)

Results of the ACID test must describe how the requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

The TPC Benchmark C standard specification defines a set of transaction processing system properties that a System Under Test (SUT) must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation and Durability (ACID). The following subsections will define each of these properties and describe the series of tests that were performed by HP to demonstrate that the properties were met.

All of the specified ACID tests were successfully performed on the HP Integrity Superdome. A fully scaled database was used for all the durability tests.

3.2 Atomicity Tests

The system under test (SUT) must guarantee that transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations have any effects on the data.

3.2.1 COMMIT Transaction

The following steps were followed to demonstrate the COMMIT property of Atomicity:

A row was randomly selected from the Warehouse, District and Customer tables, and the present balances noted. The standard payment transaction was started against the above identifiers using a known amount. The transaction was committed and the rows were verified to contain the correct updated balances.

3.2.2 ROLLBACK Transaction

The following steps were followed to demonstrate the COMMIT property of Atomicity:

A row was randomly selected from the Warehouse, District and Customer tables, and the present balances noted. The standard payment transaction was started against the above identifiers using a known amount. The transaction was rolled back and the rows were verified to contain the original balances.

3.3 Consistency Tests

Consistency is the property of the application that requires any execution of the transaction to take the database from one consistent state to another, assuming that the database is initially in a consistent state.

Consistency conditions 1 through 4 were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests. A performance run was executed at rated speed. The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

3.4 Isolation Tests

Operations of concurrent transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

This property is commonly called serializability. Sufficient conditions must be enabled at either the system or application level to ensure serializability of transactions under any mix of arbitrary transactions.

We ran a total of nine isolation tests. Seven of these tests are detailed in the TPC-C specification (clause 3.4.2.1to 3.4.2.7). The additional two are to fully comply with the isolation requirements that are not directly specified in the TPC-C specification. These two tests are known as Phantom Protection One and Two. They demonstrate that the applications are protected from phantom inserts.

3.5 Durability Tests

The tested system must guarantee the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in clause 3.5.3.1, 3.5.3.2, and 3.5.3.3.

Three types of failures were tested to ensure the durability of the database: Loss of Data, Loss of Log, and Loss of System/Memory. All test were performed on the full scale database..

3.5.1 Loss of Data

The standard driving mechanism was used to generate the transaction load of slightly more than 99,350 users for the test (10% of full load). To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The database was backed up using SQLServer backup facilities.
2. A sum of D_NEXT_O_ID was taken.
3. Slightly more than 99,350 (10%) users were logged in to the database and ran transactions.
4. After 5 minutes, one data disk drive was removed. Errors were noted on both the SQL log, OS log, and the RTE log.
5. The RTE monitor was used to verify that no users were lost.
6. The RTE was shutdown and a success file was created.
7. The database log was backed up to disc.
8. SQL was shut down, the disc re-inserted and the RAID0 volume recovered.
9. The database was restored from the original backup that was restored before the run, specifying recovery NOT be done after the restore.
10. The log was restored with recovery, effectively rolling forward all successful transactions from the run.
11. Transaction were exported from the success file. 6 New Orders were chosen at random and verified to exist in the database.

3.5.2 Loss of System / Memory and loss of Log

This was demonstrated on the full database with 112000 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 993,500 users for this test. To demonstrate recovery the following steps were followed:

1. The full database was used.
2. A sum of D_NEXT_O_ID was taken.
3. 993,500 users were logged in to the database and ran transactions.
4. Rampup was performed until the TPMC rate was 90% of the reported rate.
5. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
6. After another 5 minutes, power to the system was cut off.
7. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
8. The RTE was stopped.
9. The server machine was powered on again and rebooted.
10. Microsoft SQL Server was restarted and performed an automatic recovery.
11. A new count of D_NEXT_O_ID was taken.
12. This number was compared with the number of new orders reported by the RTE.
13. Samples were taken of the RTE log and verified against the database.

Chapter 4 Scaling and Database Population

4.1 Database Layout

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems.

The measured (tested) and priced systems used 60 HP Smart Array 6404 Disk Arrays. These cards were distributed as evenly as possible among the 4 PCI chassis. The 4 MSA1000 disk arrays that held the database log were connected to Emulex 1050 FC HBAs.

The measured database configuration used a total of 1742 disks, which included 1064 HP 36GB 15KRPM and 616 18GB 15KRPM drives for data, 56 HP 146GB, 10krpm U320 drives for log, two 18GB mirrored pairs for the operating system, and 1 36GB mirrored pair for utility. Part of the space on each of the 60 database disk arrays was configured as 1 RAID0 volume over 28 36GB or 18GB drives. Each volume held 2 partitions, one for the CS filegroup where the Customer and Stock tables were stored and one partition for MISC filegroup where all other tables were stored. The remainder of the disc space on each of the 60 Arrays was configured as a RAID5 volume over all 28 36 GB or 18GB drives. Each volume had 2 partitions, containaining backups of the data base.

Table 2a shows the complete data distribution. (Note for ease of illustration, all arrays are shown to be made up of 36GB discs, but the measured configuration consisted of 22 arrays of 18GB disks, and 38 arrays of 36GB disks. The partition sizes and SQL file sizes were the same across all arrays).

The database log drive storage was located on 4 HP MSA1000's. Each MSA1000 held 14 146MB Ultra320 SCSI hard drives. The MSA1000 disk arrays were connected to the Integrity Superdome using 4 Emulex 1050 Fibrechannel HBA's. The disks were configured as RAID 1+0, and two battery backed up disk array controller caches were enabled and mirrored on each MSA1000 (90% write, 10% read). . The 4 Arrays were then software striped using Windows Dynamic Disk Management. Not all the space visible to the OS on the 4 log arrays were configured for the log in SQL, but could have been if needed. Table 2b shows the log distribution.

Table 2a: Data Distribution

SmartArray 6404 HBA #1a			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 1	
SCSI ID	Channels		Partitions (RAID 0):162GB(18GBDisk),473GB(36GB) Disk	
	0	1	0	1
0	36GB	36GB	G:\MNT\CS1 Raw 110.43 GB	G:\MNT|MISC1 Raw 61.20 GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		
SmartArray 6404 HBA #1b			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 2	
SCSI ID	Channels		Partitions (RAID 5):302GB(18GB Disk) 458GB (36GB Disk)	
	2	3	0	1
0	36GB	36GB	G:\MNT\BACKUP\1\ NTFS Volume Backup1 150 GB	G:\MNT\BACKUP2\1\ NTFS Volume Backup1 150GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		

SmartArray 6404 HBA #2b			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 3	
SCSI ID	Channels		Partitions (RAID 0):162GB(18GBDisk),473GB(36GB) Disk	
	0	1	0	1
0	36GB	36GB	G:\MNT\CS2 Raw 110.43B	G:\MNT|MISC2 Raw 61.20GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		
SmartArray 6404 HBA #2b			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 4	
SCSI ID	Channels		Partitions (RAID 5):302GB(18GB Disk) 458GB (36GB Disk)	
	2	3	0	1
0	36GB	36GB	G:\MNT\BACKUP\2 NTFS Volume Backup2 150GB	G:\MNT\BACKUP2\2 NTFS Volume Backup2 150GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		

…..

….

SmartArray 6404 HBA #30a			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 23	
SCSI ID	Channels		Partitions (RAID 0):162GB(18GBDisk),473GB(36GB) Disk	
	0	1	0	1
0	36GB	36GB	G:\MNT\CS60 Raw 110.43GB	G:\MNT|MISC60 Raw 61.20GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		
SmartArray 6404 HBA #30b			WINDOWS.NET DISK ADMIN	
2 StorageWorks			DISK 24	
SCSI ID	Channels		Partitions (RAID 5):302GB(18GB Disk) 458GB (36GB Disk)	
	2	3	0	1
0	36GB	36GB	G:\MNT\BACKUP\60 NTFS Volume Backup24 150GB	G:\MNT\BACKUP2\60 NTFS Volume Backup24 150GB
1	36GB	36GB		
2	36GB	36GB		
3	36GB	36GB		
4	36GB	36GB		
5	36GB	36GB		
6	36GB	36GB		
8	36GB	36GB		
9	36GB	36GB		
10	36GB	36GB		
11	36GB	36GB		
12	36GB	36GB		
13	36GB	36GB		
14	36GB	36GB		

Table 2b: Log Distribution

Emulex 1050 Fiber Adapter #1		WINDOWS.NET DISK ADMIN	
1 MSA1000		DISK 1 (Dynamic)	
SCSI ID	Channels	Partitions (RAID 1+0 980 GB)	
	0	0	1
0	146GB	(no drv ltr) Raw 10 MB (for proper alignment)	L: Raw 956.99 GB (NT Software Striped)
1	146GB		
2	146GB		
3	146GB		
4	146GB		
5	146GB		
6	146GB		
8	146GB		
9	146GB		
10	146GB		
11	146GB		
12	146GB		
13	146GB		
14	146GB		
Emulex 1050 Fiber Adapter #2		WINDOWS.NET DISK ADMIN	
1 MSA1000		DISK 2 (Dynamic)	
SCSI ID	Channels	Partitions (RAID 1+0 980GB)	
	0	0	1
0	146GB	(no drv ltr) Raw 10MB (for proper alignment)	L: Raw 956.99 GB (NT Software Striped)
1	146GB		
2	146GB		
3	146GB		
4	146GB		
5	146GB		
6	146GB		
8	146GB		
9	146GB		
10	146GB		
11	146GB		
12	146GB		
13	146GB		
14	146GB		

Emulex 1050 Fiber Adapter #2		WINDOWS.NET DISK ADMIN	
1 MSA1000		DISK 3 (Dynamic)	
SCSI ID	Channels	Partitions (RAID 1+0 980GB)	
	0	0	1
0	146GB	(no drv ltr) Raw 10 MB (for proper alignment)	L: Raw 956.99 GB (NT Software Striped)
1	146GB		
2	146GB		
3	146GB		
4	146GB		
5	146GB		
6	146GB		
8	146GB		
9	146GB		
10	146GB		
11	146GB		
12	146GB		
13	146GB		
14	146GB		
Emulex 1050 Fiber Adapter #4		WINDOWS.NET DISK ADMIN	
1 MSA1000		DISK 4 (Dynamic)	
SCSI ID	Channels	Partitions (RAID 1+0 980GB)	
	0	0	1
0	146GB	(no drv ltr) Raw 10MB (for proper alignment)	L: Raw 956.99 GB (NT Software Striped)
1	146GB		
2	146GB		
3	146GB		
4	146GB		
5	146GB		
6	146GB		
8	146GB		
9	146GB		
10	146GB		
11	146GB		
12	146GB		
13	146GB		
14	146GB		

4.2 Initial Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted, the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

Table 3 shows the cardinality of the various tables.

Table 3: Table Cardinality

Table	Occurrences
Warehouse	112,000
District	1,120,000
Customer	3,360,000,000
History	3,360,000,000
Orders	3,360,000,000
New Orders	1,008,000,000
Order Line	840,041,944
Stock	11,200,000,000
Item	100,000

No rows were deleted for the benchmark runs.

4.3 60 Day Space

Details of the 60 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables must be disclosed.

4.3.1 Transaction Log Space Requirements

To calculate the space required to sustain the database log for 8 hours of growth at steady state, the following steps were followed:

1. The free space on the logfile was queried using **dbcc sqlperf(logspace)**.
2. Transactions were run against the database with a full load of users.
3. The free space was again queried using **dbcc sqlperf(logspace)**.
4. The space used was calculated as the difference between the first and second query.
5. The number of NEW-ORDERS was verified from an RTE report covering the entire run.
6. The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
7. The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The result of the above steps yielded a requirement of 646 GB to sustain the log for 8 hours. Space available for the transaction log was 820 GB indicating that enough storage was configured to hold 8 hours of growth.

The same methodology was used to calculate the growth requirements for the other dynamic tables Order, Order-Line and History. The details of the 60ay growth calculation are shown in Appendix D.

4.4 Type of Database Used

A statement must be provided that describes:

1. *The data model implemented by the DBMS used (e.g., relational, network, hierarchical)*

2. *The database interface (e.g., embedded, call level) and access language (e.g., SQL, DL/1, COBOL read/write) used to implement the TPC-C transactions. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.*

Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 is a relational DBMS.

The interface was SQL Server stored procedures accessed with library calls embedded in C code.

4.5 Database Mapping

The mapping of database partitions and replications must be described.

The database was divided into 2 file groups misc_fg and cs_fg. misc_fg consists of 60 partitions at 110430 MB each and misc_fg consist of 60 partitions at 61200 MB each as shown in the createdb.sql. The log was configured with 20,000 MB at database creation, and was expanded to 1,500,000 MB after data base creation and load.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 1,231,433
Price per TpmC®: $4.82 USD

5.2 Response Times

Ninetieth percentile, maximum and average response times must be reported for all transactions types as well as for the menu response time.

Table 3 shows the response times for all transaction types.

Table 4: Transaction Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.40s	0.92s	13.29s
Payment	0.37s	0.89s	12.23s
Order-Status	0.42s	0.95s	11.79s
Delivery (interactive portion)	0.20s	0.42s	10.65s
Delivery (deferred portion)	0.14s	0.19s	10.72s
Stock-Level	0.52s	1.07s	12.04s
Menu	0.20s	0.42s	10.65s

5.3 Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

Tables 4 and 5 show the key times and think times for all transaction types.

Table 5: Transaction Key Times

Keying Times	Minimum	Average	Maximum
New Order	18.00	18.01s	20.78s
Payment	3.00	3.02s	5.28s
Order Status	3.00	3.02s	5.23s
Interactive Delivery	3.00	3.02s	5.17s
Stock Level	3.00	3.02s	5.27s

Table 6: Transaction Think Times

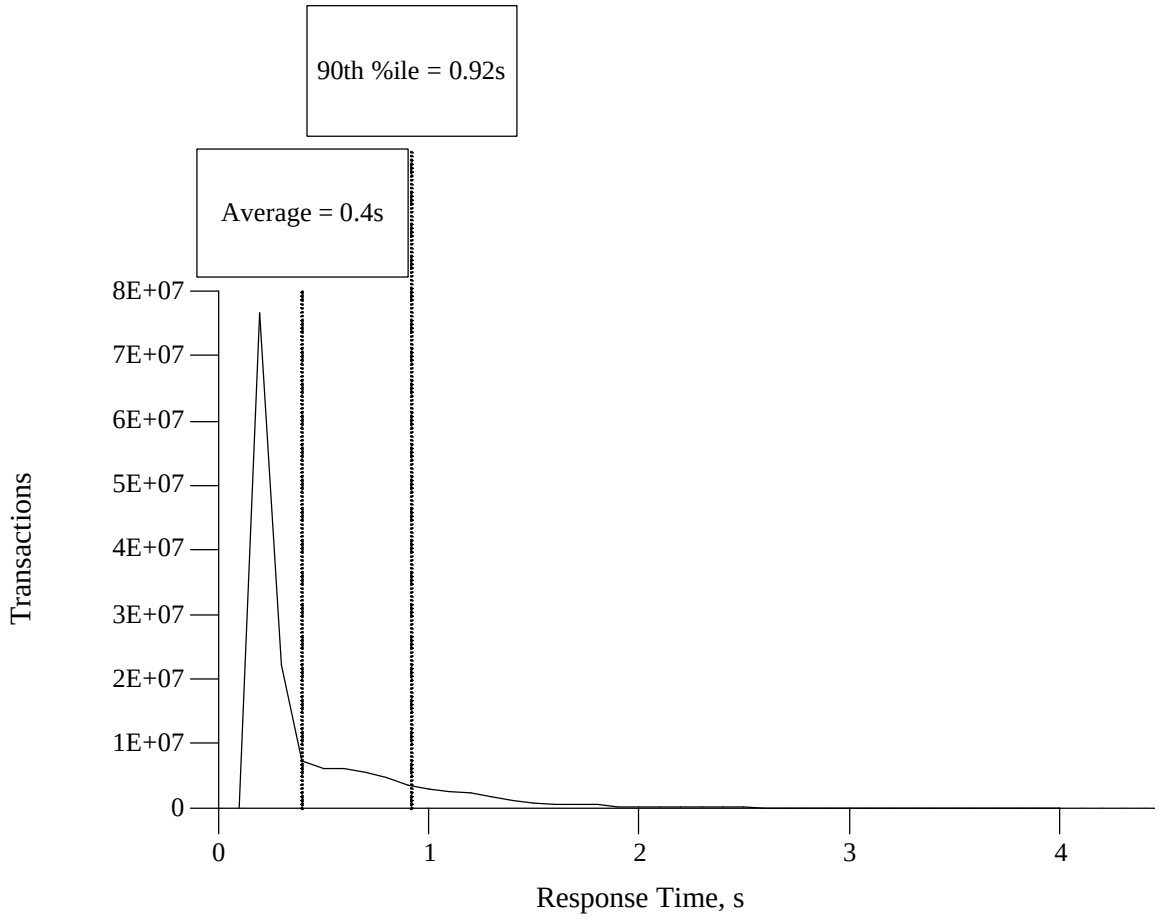
Think Times	Minimum	Average	Maximum
New Order	0	12.07s	121.28s
Payment	0	12.06s	120.91s
Order Status	0	10.06s	100.53s
Interactive Delivery	0	5.07s	50.53s
Stock Level	0	5.06s	50.53s

5.4 Response Time Frequency

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type. Keying Time frequency distribution curves (see Clause5.6.4) must be reported for each transaction type. A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

5.4.1 New Order Response Time

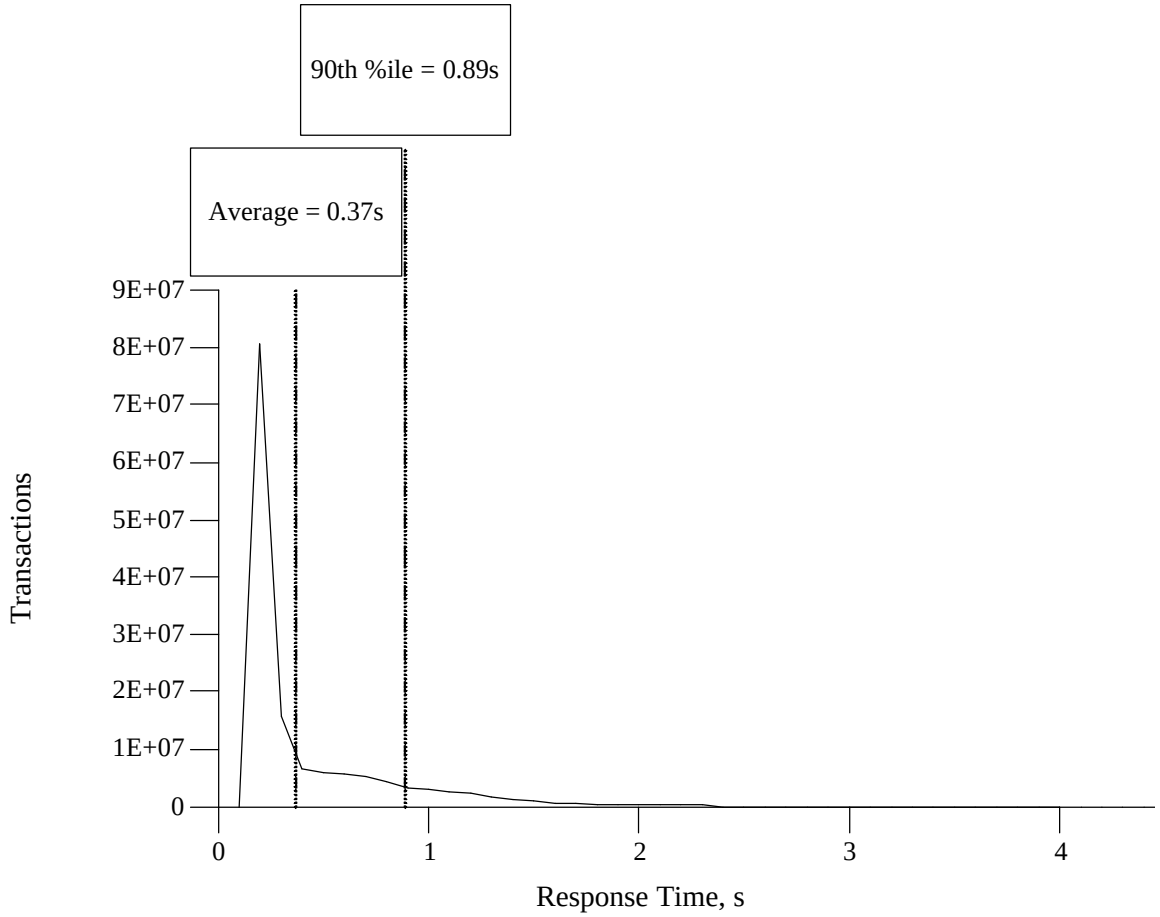
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

5.4.2 Payment Response Time Distribution

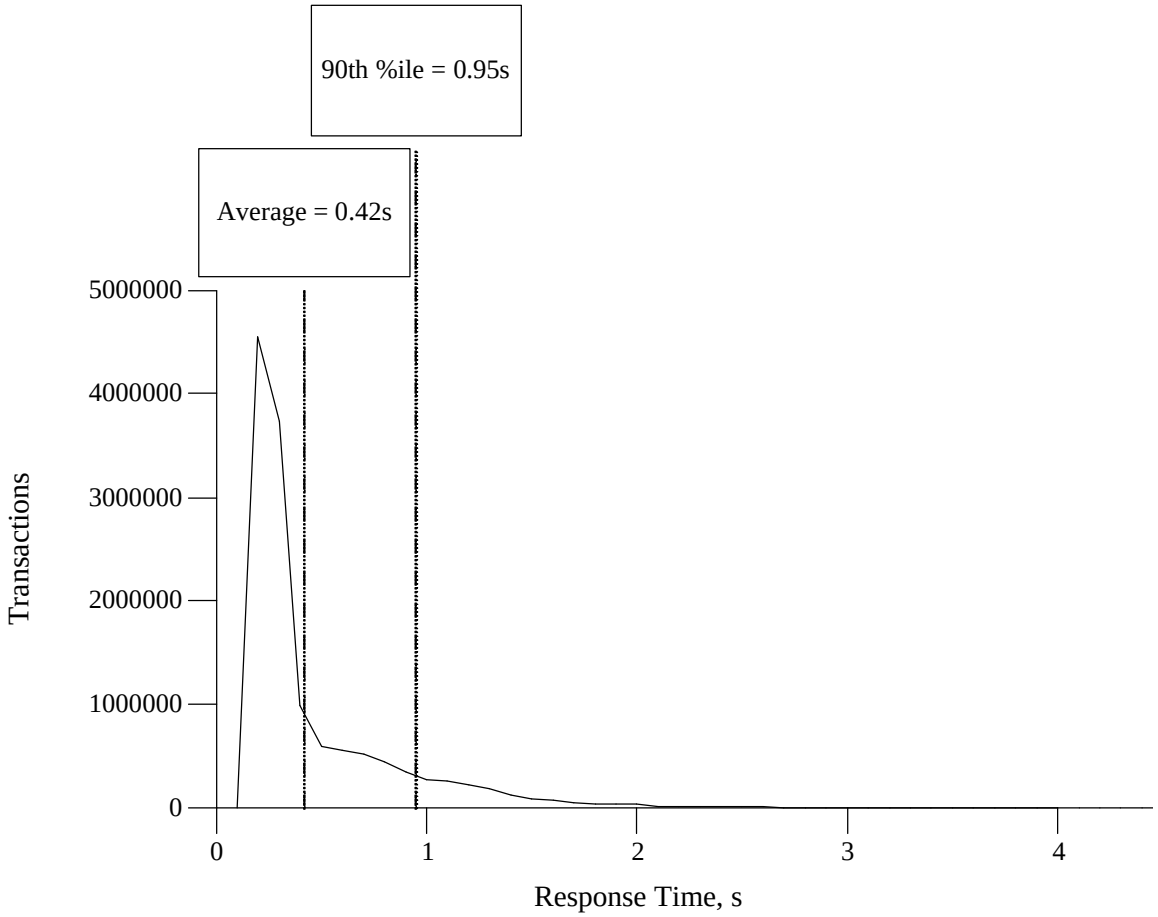
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

5.4.3 Order Status Response Time

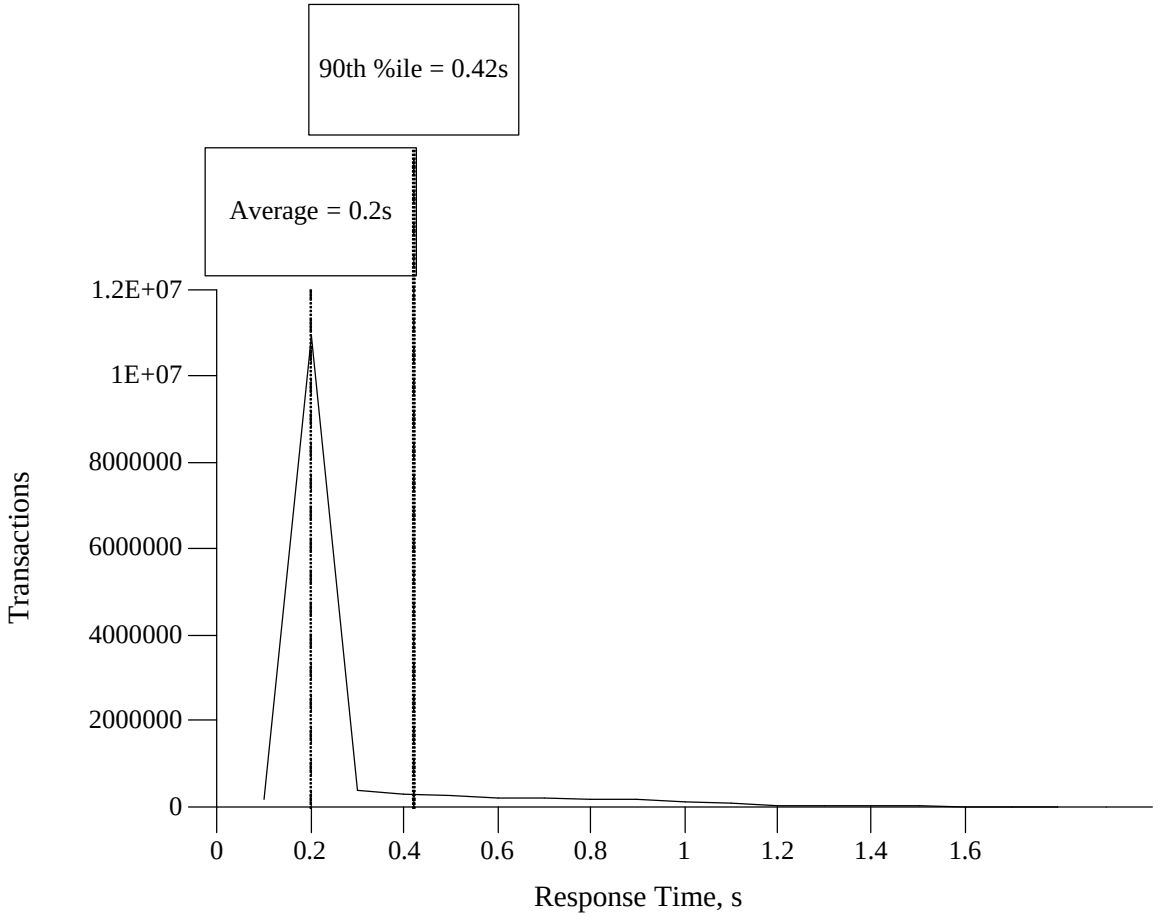
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

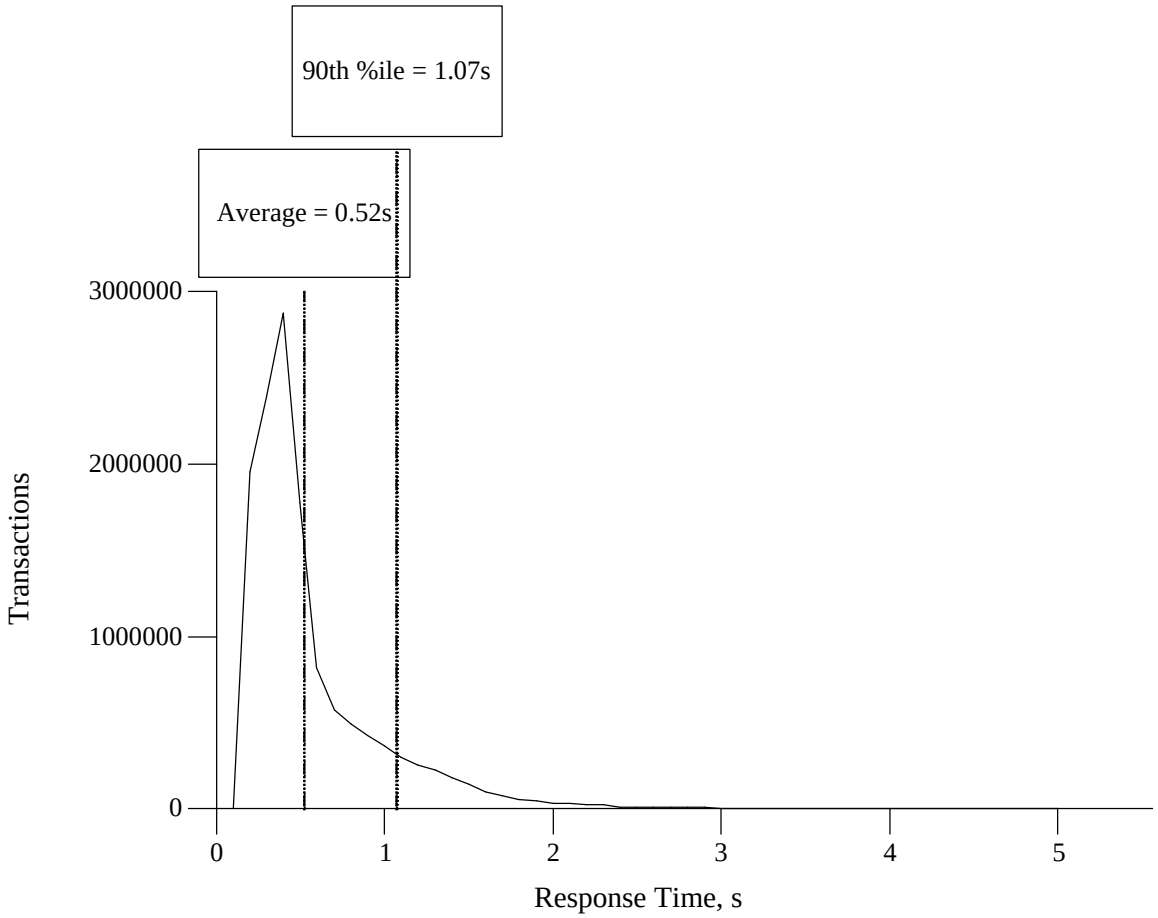
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

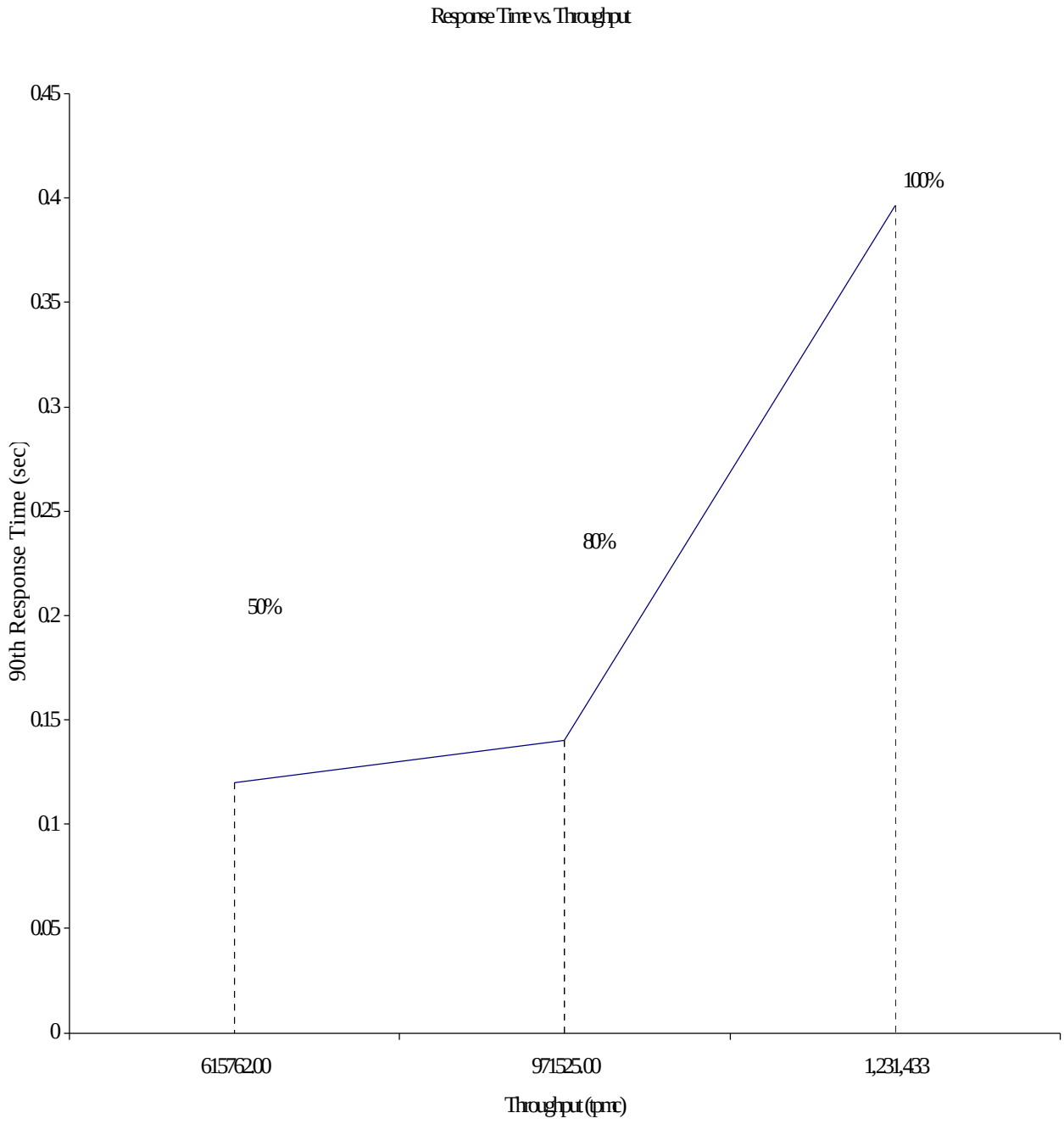
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

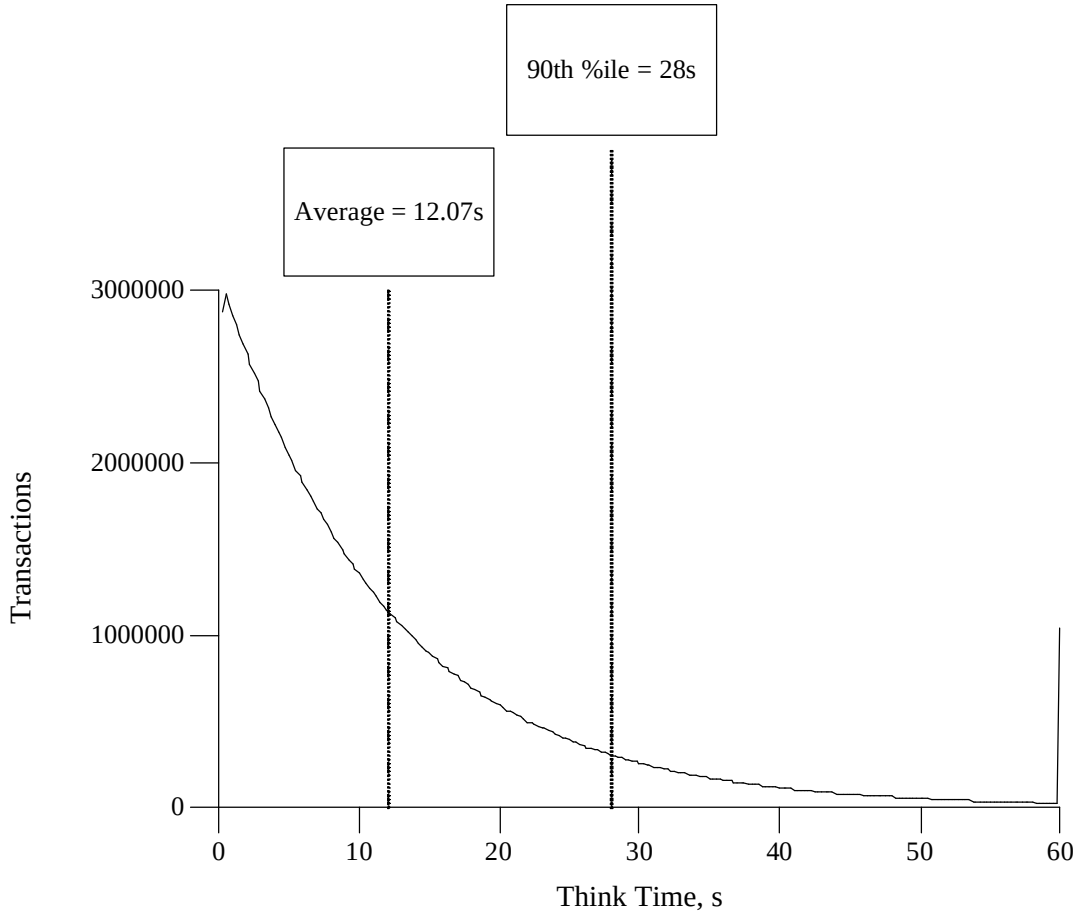
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

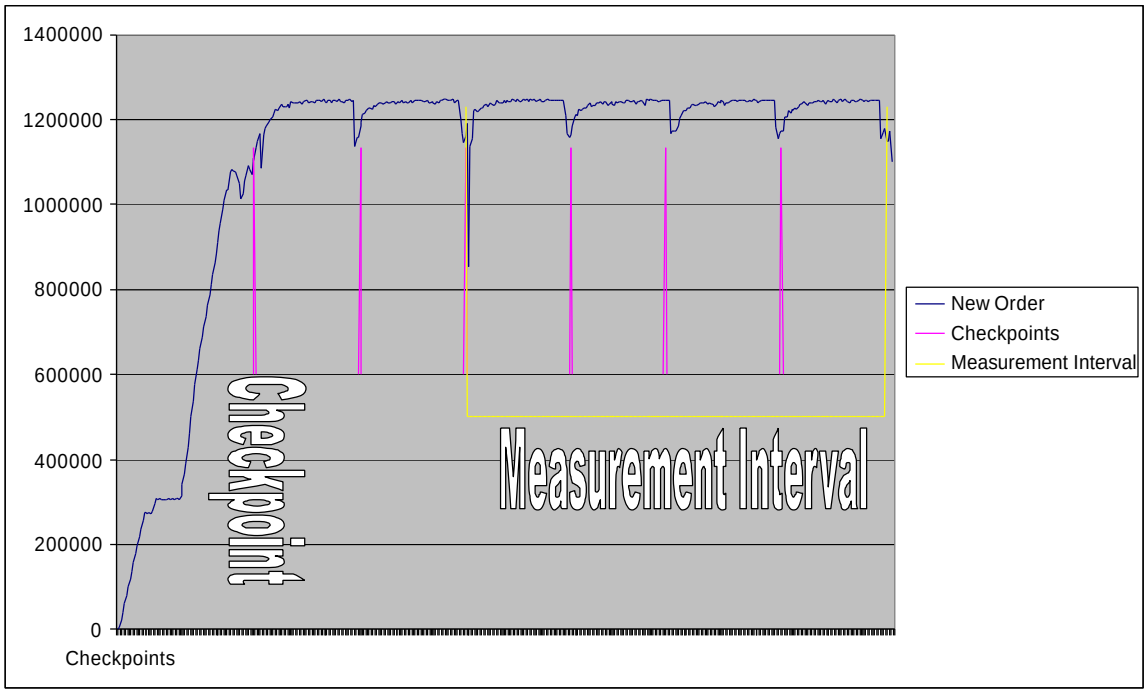
Figure 9: New Order Think Time Distribution



Think time frequency distribution for New Order transaction

5.4.8 Throughput Versus Time Distribution

Figure 10: New Order Throughput versus Time



5.5 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be disclosed.

The transaction throughput rate (tpmC®) and response time were relatively constant after the initial 'ramp up' period. The throughput and response time behaviors were determined by examining data reported for each interval over the duration of the benchmark. The corresponding graph is in Figure 10.

5.6 Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.), actually occurred during the measurement interval must be reported.

The RTEs generated the required input data to choose a transaction from the menu. This data was timestamped. The menu response time for the requested transaction was verified and timestamped in the RTE log files. The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the HTTP request to the client. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction and was logged in the RTE log. The RTE then waited the required think time interval before repeating the process and starting another transaction.

5.6.1 Checkpoint

The checkpoint mechanism is an automatic means for guaranteeing that completed transactions are regularly written from SQL Server's disk cache to the database device. A checkpoint writes all "dirty pages"-cached pages that have been modified since the last checkpoint-to the database device.

5.6.2 Checkpoint Conditions

There are two types of checkpoints:

1. Checkpoints that are executed automatically by SQL Server.

2. Checkpoints that are forced by database owners with the CHECKPOINT statement.

Forcing dirty pages onto the database device means that all completed transactions are written out. By causing all completed transactions to be written out, the checkpoint shortens the time it takes to recover, since the database pages are current and there are no transactions that need to be rolled forward.

5.6.3 Checkpoint Implementation

A Windows command script was issued to start manual checkpoints back to back. The "CHECKPOINT 1700" syntax in Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 was used to force the checkpoints to an interval of 28 minutes, 20 seconds. The checkpoints were affinitized to a 1 processor SoftNuma node using tcp connection affinity. The script was run on the database server. By setting the TRACE FLAG #3502, SQL Server logged the checkpoint beginning and ending time in the ERRORLOG file.

At each checkpoint, Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 wrote to disk all memory pages that had been updated but not yet physically written to disk. Upon completion of the checkpoint, Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 wrote a special record to the recovery log to indicate that all disk operations had been satisfied to this point. The positioning of the checkpoint was verified to be clear of the guard zones.

5.7 Measurement Period Duration

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC®) must be included.

The measurement interval was 120 minutes.

5.8 Regulation of Transaction Mix

The method of regulation of the transaction mix (e.g. card decks, or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The weighted average method of *Clause 5.2.4.1* was used. The weights were not adjusted during the run.

5.9 Transaction Mix

The percentage of the total mix for each transaction type must be disclosed.

Table 7: Transaction Mix

Type	Percentage
New Order	44.97%
Payment	43.01%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

5.10 Transaction Statistics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order-lines entered per New-Order transaction must be disclosed. The percentage of selections made by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 1 contains the required items.

5.11 Checkpoint Count and Location

The number of checkpoints in the measurement interval, the time in seconds from the start of the measurement interval to the first checkpoint, and the Checkpoint Interval must be disclosed.

The measurement interval is 120 minutes. There are 4 checkpoints within the measurement interval and 2 checkpoint before the measurement interval.

Table 8: Measurement Interval and Checkpoints

Event	From	To
Measured Interval	20:37:56	22:37:56
Checkpoint	19:38:00	20:06:20
Checkpoint	20:07:58	20:36:18
Checkpoint	20:37:56	21:06:16
Checkpoint	21:07:54	21:36:14
Checkpoint	21:37:52	22:06:12
Checkpoint	22:07:50	22:36:10

Chapter 6 SUT, Driver and Communications Definition

6.1 RTE Description

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of that input (e.g., scripts) to the RTE had been used. The RTE input parameters, code fragments, functions, et cetera used to generate each transaction input filed must be disclosed.

The RTE used is Microsoft BenchCraft and is commercially available. The RTE input parameters are listed in Appendix C – Tunable Parameters.

6.2 Emulated Components

It must be demonstrated that the functionality and performance of the components being used in the Driver System are equivalent to that of the priced system.

No components were emulated.

6.3 Functional Diagram

A complete functional diagram of the hardware and software of the benchmark configuration including the driver must be provided. the sponsor must list all hardware and software functionality of the driver and its interface to the SUT.

Functional diagrams of the measured and priced systems are included in the "General Items" section at the beginning of this report.

6.4 Networks

The network configuration of both the tested and proposed services which are being represented and a thorough explanation of exactly which parts are being replaced with the Driver System must be disclosed.

The "General Items" section includes diagrams of the network configurations of the benchmark and configured systems, and represent the driver connected via LAN.

The bandwidth of the networks used in the tested/priced configurations must be disclosed.

A Gigabit network was used between the RTEs and the clients, a 2GBps VIA/SAN connection was used between the clients and the database server.

6.5 Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed.

The configuration does not require any operator to sustain eight hours of the reported throughput.

Chapter 7 Pricing

7.1 System Pricing

A detailed list of hardware and software used in the priced system must be reported. Each separately orderable item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery data. If package pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source and effective date(s) of price(s) must also be reported.

The total 3 year price of the entire configuration must be reported, including: hardware, software, maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The details of the hardware, software and maintenance components of this system are reported in the front of this report as part of the executive summary.

All 3rd party quotations are included at the end of this report in Appendix E.

7.2 General Availability, Throughput and Price Performance

The committed delivery date for general availability (availability date) of products used in the price calculation must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

A statement of the measured tpmC as well as the respective calculations for the 5-year pricing, price/performance and the availability date must be included.

Table 9: Throughput, Price Performance and Availability

Maximum qualified throughput:	1,231,433 tpmC
Price per tpmC:	$4.82 USD per tpmC
Availability:	May 5, 2006

7.3 Country Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country specific priced item configuration. Country specific pricing is subject to Clause 7.1.7.

The system is being priced for the United States of America.

7.4 Usage Pricing

For any usage pricing, the sponsor must disclose: Usage level at which the component was priced, a statement of the company policy allowing such pricing.

The component pricing based on usage is shown below:

- 64 Microsoft SQL Server 2005 Enterprise Edition 64-bit Service Pack 1 per-processor licenses.
- Microsoft Windows Server 2003, Datacenter edition (64-bit)
- 31 Microsoft Windows 2000 Server licenses.
- 1 Microsoft Visual C++ 32bit Edition.
- 3 year support for hardware components

Chapter 8 Audit

8.1 Auditor's Information

The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.

The test methodology and results of this TPC Benchmark C were audited by:

Performance Metrics
PO Box 984
140 Klamath Blvd
Klamath, CA 95548
(707) 482-0523
Fax (707) 482-0575

The auditor was Lorna Livingtree.
Requests for this Full Disclosure Report (FDR) should sent to:

Hewlett-Packard Company
WIE
10955 Tantau Avenue
Cupertino, CA 95014-0770 USA

A copy of the attestation letter received from the auditor follows:

5

MR. JASON GOERTZ
ISS Redmond Performance
Hewlett-Packard Company
14475 NE 24th St.
Bellevue, WA 98007

I have verified the TPC Benchmark™ C client/server for the following configuration:

Platform:	HP Integrity Superdome
Database Manager:	Microsoft SQL Server 2005 Enterprise Edition 64 bit
Operating System:	Microsoft Windows Server 2003, Datacenter Edition 64 bit
Transaction Monitor:	Microsoft COM+

Server:				
CPU's	Memory	Disks (total)	90% Response	TpmC
64 Intel Itanium 2 @ 1.6 Ghz	1,024 GB	592 @ 18 GB 1094 @ 36 GB 56 @ 146 GB	0.92	**1,231,433**
31 Clients: ProLiant DL140 each with:				
2 Pentium Xeon @ 3.06 Ghz	Main: 1.75 GB	1 @ 80 GB	Na	Na

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark. The following attributes of the benchmark were given special attention:

- The transactions were correctly implemented.
- The database files were properly sized and populated for 112,000 warehouses, of which 99,350 were active during the measured interval.
- Inactive warehouses were verified to be unchanged during the performance run.
- The ACID properties were successfully demonstrated.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the measured system.
- Eight hours of growth space for the dynamic tables was present on the measured system.
- The data for the 60 day space calculation was verified.
- The Measured cycle times were confirmed to have the correct response time delays.
- There were 993,500 user contexts present on the system.
- Each emulated users started with a different random number seed.
- The NURand constants used for C_last load and run were 123 and 233 respectively.
- The steady state portion of the test was 120 minutes.
- More than one checkpoint was taken after steady state and before the measured interval.
- Four checkpoints were contained completely inside the measured interval, and additional checkpoints continued after the measured interval closed.
- Checkpoint interval was verified to be less than 30 minutes.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes:
The measured configuration had 36 full arrays populated with 28 of the priced disks. The rest of the configuration was running older, obsolete disks. The priced disks had better technical specifications and capacities than the obsolete disks. Performance data was collected on a TPC-C performance run and proved the

priced disks meet the substitution requirements. This substitution is compliant with the disk substitution rules.

Sincerely,

Lorna Livingtree
Auditor

Appendix A Source Code

Isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

        GetExtensionVersion     @1
        HttpExtensionProc       @2
        TerminateExtension      @3
```

Isapi_dll/src/tpcc.h

```
/*      FILE:                   TPCC.H
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:    Header file for ISAPI TPCC.DLL, defines
structures and functions used in the isapi tpcc.dll.
 *
 */

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
        101
#define _APS_NEXT_COMMAND_VALUE
                40001
#define _APS_NEXT_CONTROL_VALUE
                1000
#define _APS_NEXT_SYMED_VALUE
                101

#define TP_MAX_RETRIES
                        50

//note that the welcome form must be processed first as terminal ids
assigned here, once the
//terminal id is assigned then the forms can be processed in any order.
#define WELCOME_FORM
                1               //beginning form
no term id assigned, form id
#define MAIN_MENU_FORM
                        2
        //term id assigned main menu form id
#define NEW_ORDER_FORM
                        3
        //new order form id
#define PAYMENT_FORM
                4               //payment form
id
#define DELIVERY_FORM
                5               //delivery form id
#define ORDER_STATUS_FORM
                6               //order status id
#define STOCK_LEVEL_FORM
                7               //stock level form
id

//This macro is used to prevent the compiler error unused formal
parameter
#define UNUSEDPARAM(x) (x = x)


//This structure defines the data necessary to keep distinct for each
terminal or client connection.
typedef struct _CLIENTDATA
{
        int                     iNextFree;
                        //index of next free element or
-1 if this entry in use.
        int                     w_id;
                        //warehouse id
assigned at welcome form
        int                     d_id;
                        //district id
assigned at welcome form

        int                     iSyncId;
                        //syncronization id
        int                     iTickCount;
                        //time of last access;

        CTPCC_BASE *pTxn;

} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational interface for terminal
id support

typedef struct _TERM
{
        int                     iNumEntries;
                                //total allocated
terminal array entries
        int                     iFreeList;

        //next available terminal array element or -1 if none
        int                     iMasterSyncId;

        //syncronization id
        CLIENTDATA *pClientData;
                //pointer to allocated client data
} TERM;

typedef TERM *PTERM;
                        //pointer to terminal structure
type


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,

        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
```

```
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,

                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,

        ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

                CWEBCLNT_ERR(WEBERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };
                CWEBCLNT_ERR(WEBERROR Err, char
*szTextDetail, DWORD dwSystemErr)
                        m_Error = Err;
                        m_szTextDetail = new
char[strlen(szTextDetail)+1];
                        strcpy( m_szTextDetail,
szTextDetail );
                        m_SystemErr = dwSystemErr;
                        m_szErrorText = NULL;
                };

                ~CWEBCLNT_ERR()
                {
                        if (m_szTextDetail != NULL)
                                delete []
m_szTextDetail;
                        if (m_szErrorText != NULL)
                                delete []
m_szErrorText;
                };

                WEBERROR  m_Error;
                char                    *m_szTextDetail;
        //
                char                    *m_szErrorText;
                DWORD                   m_SystemErr;

                int ErrorType() {return
ERR_TYPE_WEBDLL;};
                int ErrorNum() {return m_Error;};
                char *ErrorText();

};

//These constants have already been defined in engstut.h, but since we
do
//not want to include it in the delisrv executable
#define     TXN_EVENT_START                 2
#define     TXN_EVENT_STOP                  4
#define TXN_EVENT_WARNING       6
        //used to record a warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int
*pCmd, int *pFormId, int *pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int
iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int
iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int
iMax, WEBERROR err);
int  GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR
NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int  TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData,
BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
                *pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
```

Isapi_dll/src/tpcc.rc

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

//////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

//////////////////////////////////////////////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

#ifndef _MAC
//////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 0,4,0,0
 PRODUCTVERSION 0,4,0,0
 FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x40004L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C HTML DLL Server (DBLIB)\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "TPC-C HTML DLL Server (DBLIB)\0"
            VALUE "FileVersion", "0, 4, 0, 0\0"
            VALUE "InternalName", "tpcc\0"
            VALUE "LegalCopyright", "Copyright © 1997\0"
            VALUE "OriginalFilename", "tpcc.dll\0"
            VALUE "ProductName", "Microsoft tpcc\0"
            VALUE "ProductVersion", "0, 4, 0, 0\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


#ifdef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include ""afxres.h""\r\n"
    "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "\r\n"
    "\0"
END

#endif    // APSTUDIO_INVOKED


//////////////////////////////////////////////////////////////////////////////
//
// Dialog
//

IDD_DIALOG1 DIALOG DISCARDABLE  0, 0, 186, 95
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "Dialog"
FONT 8, "MS Sans Serif"
BEGIN
    DEFPUSHBUTTON   "OK",IDOK,129,7,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,129,24,50,14
END

//////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif    // APSTUDIO_INVOKED

#endif    // English (U.S.) resources
//////////////////////////////////////////////////////////////////////////////



#ifndef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
//////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

Isapi_dll/src/tpcc.cpp

```
/*            FILE:                    TPCC.C
 *                                                    Microsoft TPC-C
Kit Ver. 4.20.000
 *                                                    Copyright
Microsoft, 1999
 *          All Rights Reserved
 *
 *                                                    Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *            PURPOSE:    Main module for TPCC.DLL which is an ISAPI
service dll.
 *            Contact:    Charles Levine (clevine@microsoft.com)
 *
```

```
 *  Change history:
 *                          4.20.000 - reworked error handling; added
options for COM and Encina txn monitors
 */

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

#include "..\..\common\src\trans.h"                     //tpckit
transaction header contains defination of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\common\txnlog\include\rtetime.h"
#include "..\..\common\txnlog\include\spinlock.h"
#include "..\..\common\txnlog\include\txnlog.h"

// Database layer includes
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  //
DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    //
ODBC implementation of TPC-C txns

// Txn monitor layer includes
#include "..\..\tm_com_dll\src\tpcc_com.h"
        // COM Services implementation on TPC-C txns
#include "..\..\tm_tuxedo_dll\src\tpcc_tux.h"                   //
interface to Tuxedo libraries
#include "..\..\tm_encina_dll\src\tpcc_enc.h"                   //
interface to Encina libraries

#include "httpext.h"
        //ISAPI DLL information header
#include "tpcc.h"
        //this dlls specific structure, value e.t. header.

#define     LEN_ERR_STRING          256

// defines for Make<Txn>Form calls to distinguish input and output
flavors
#define     OUTPUT_FORM             0
#define INPUT_FORM      1

char
        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

//Terminal client id structure
TERM        Term = { 0, 0, 0, NULL };

// The WEBCLIENT_VERSION string specifies the version level of this
web client interface.
// The RTE must be synchronized with the interface level on login,
otherwise the login
// will fail.  This is a sanity check to catch problems resulting from
mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "420"

static      CRITICAL_SECTION
        TermCriticalSection;

static  HINSTANCE hLibInstanceTm = NULL;
static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB        *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;
TYPE_CTPCC_TUXEDO     *pCTPCC_TUXEDO_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM                  *pCTPCC_COM_new;

// For deferred Delivery txns:

CTxnLog
        *txnDelilog = NULL;
        //used to log delivery transaction information

HANDLE
        hWorkerSemaphore        = INVALID_HANDLE_VALUE;
HANDLE
        hDoneEvent                      =
INVALID_HANDLE_VALUE;
HANDLE
        *pDeliHandles                   = NULL;

// configuration settings from registry
TPCCREGISTRYDATA                Reg;

DWORD
        dwNumDeliveryThreads = 4;
CRITICAL_SECTION                DelBuffCriticalSection;
        //critical section for delivery transactions cache
DELIVERY_TRANSACTION *pDelBuff                          =
NULL;
DWORD
        dwDelBuffSize                   = 100;
        // size of circular buffer for delivery txns
DWORD
        dwDelBuffFreeCount;
        // number of buffers free
DWORD
        dwDelBuffBusyIndex      = 0;                    //
index position of entry waiting to be delivered
DWORD
        dwDelBuffFreeIndex      = 0;                    //
index position of unused entry

// Critical section to synchronize connection open and close.
//
CRITICAL_SECTION hConnectCriticalSection;

#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: DllMain
 *
 * PURPOSE: This function is the entry point for the DLL.  This
implementation is based on the
 *                          fact that
DLL_PROCESS_ATTACH is only called from the inet service once.
 *
 * ARGUMENTS:       HANDLE      hModule
                    module handle
 *                                      DWORD
        ul_reason_for_call      reason for call
 *                                      LPVOID
        lpReserved                      reserved for
future use
 *
 * RETURNS:         BOOL        FALSE
                    errors occured in initialization
 *
        TRUE                                            DLL
successfully initialized
 */

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
        DWORD  i;
        char szEvent[LEN_ERR_STRING] = "\0";
        char szLogFile[128];
```

```
	char szDllName[128];

// debugging....
// DebugBreak();

	try
	{
		switch( ul_reason_for_call )
		{
			case DLL_PROCESS_ATTACH:
				{

	DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;

	GetComputerName(szMyComputerName, &dwSize);

	szMyComputerName[dwSize] = 0;
				}


	DisableThreadLibraryCalls((HMODULE)hModule);

	InitializeCriticalSection(&TermCriticalSection);

				if (
ReadTPCCRegistrySettings( &Reg ) )

	throw new CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

				dwDelBuffSize =
min( Reg.dwMaxPendingDeliveries, 10000 );  // min with 10000 as a
sanity constraint

	dwNumDeliveryThreads = min(
Reg.dwNumberOfDeliveryThreads, 100 );  // min with 100 as a sanity
constraint

				TermInit();

				// load DLL for
txn monitor
				if (Reg.eTxnMon
== TUXEDO)
				{

	strcpy( szDllName, Reg.szPath );

	strcat( szDllName, "tpcc_tuxedo.dll");

	hLibInstanceTm = LoadLibrary( szDllName );
```

```
				if
(hLibInstanceTm == NULL)
	throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

				//
get function pointer to wrapper for class constructor

	pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
				if
(pCTPCC_TUXEDO_new == NULL)

	throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
				}
				else if
(Reg.eTxnMon == ENCINA)
				{

	strcpy( szDllName, Reg.szPath );

	strcat( szDllName, "tpcc_encina.dll");

	hLibInstanceTm = LoadLibrary( szDllName );
				if
(hLibInstanceTm == NULL)

	throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
				//
get function pointer to wrapper for class constructor

	pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");

	pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
				if
(pCTPCC_ENCINA_new == NULL)

	throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
				}
				else if
(Reg.eTxnMon == COM)
				{

	strcpy( szDllName, Reg.szPath );

	strcat( szDllName, "tpcc_com.dll");
```

```
	hLibInstanceTm = LoadLibrary( szDllName );
				if
(hLibInstanceTm == NULL)

	throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

				//
get function pointer to wrapper for class constructor

	pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm,"CTPCC_COM_new");
				if
(pCTPCC_COM_new == NULL)

	throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
				}

				// load DLL for
database connection
				if ((Reg.eTxnMon
== None) || (dwNumDeliveryThreads > 0))
				{
				if
(Reg.eDB_Protocol == DBLIB)
				{

	strcpy( szDllName, Reg.szPath );

	strcat( szDllName, "tpcc_dblib.dll");

	hLibInstanceDb = LoadLibrary( szDllName );

	if (hLibInstanceDb == NULL)

		throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );


	// get function pointer to wrapper for class constructor

	pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");

	if (pCTPCC_DBLIB_new == NULL)

		throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
				}
				else
if (Reg.eDB_Protocol == ODBC)
				{
```

```
            strcpy( szDllName, Reg.szPath );

            strcat( szDllName, "tpcc_odbc.dll");

            hLibInstanceDb = LoadLibrary( szDllName );

            if (hLibInstanceDb == NULL)

                        throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );


            // get function pointer to wrapper for class constructor

            pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");

            if (pCTPCC_ODBC_new == NULL)

                        throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );


                                                }
                                    }

                                    if
(dwNumDeliveryThreads)
                                    {
                                                //
Initialize delivery delay critical section
                                                //

            InitializeCriticalSection(&hConnectCriticalSection);

                                                //
for deferred delivery txns:

            hDoneEvent = CreateEvent( NULL, TRUE /* manual reset
*/, FALSE /* initially not signalled */, NULL );

            InitializeCriticalSection(&DelBuffCriticalSection);

            hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );

            dwDelBuffFreeCount = dwDelBuffSize;


            InitJulianTime(NULL);
```

```
                                                //
create unique log file name based on delilog-yymmdd-hhmm.log
            SYSTEMTIME Time;

            GetLocalTime( &Time );

            wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-
%2.2d%2.2d-%2.2ds%2.2dms.log",

            Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute, Time.wSecond,
Time.wMilliseconds );

            txnDelilog = new CTxnLog(szLogFile, TXN_LOG_WRITE);


            //write event into txn log for START

            txnDelilog->WriteCtrlRecToLog(TXN_EVENT_START,
szMyComputerName, sizeof(szMyComputerName));

                                                //
allocate structures for delivery buffers and thread mgmt

            pDeliHandles = new HANDLE[dwNumDeliveryThreads];

            pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                                //
launch DeliveryWorkerThread to perform actual delivery txns

            for(i=0; i<dwNumDeliveryThreads; i++)
                                                {

            pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );

            if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

                        throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
                                                }
                                    }

                                    break;

                        case DLL_PROCESS_DETACH:
                                    if
(dwNumDeliveryThreads)
                                    {
                                                if
(txnDelilog != NULL)
                                                {
```

```
            //write event into txn log for STOP
            txnDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP,
szMyComputerName, sizeof(szMyComputerName));


            // This will do a clean shutdown of the delivery log file

            CTxnLog      *txnDelilogLocal = txnDelilog;

            txnDelilog= NULL;

            delete txnDelilogLocal;
                                                }


            delete [] pDeliHandles;

            delete [] pDelBuff;


            CloseHandle( hWorkerSemaphore );

            CloseHandle( hDoneEvent );

            DeleteCriticalSection(&DelBuffCriticalSection);

                                                //
Delete delivery delay critical section
                                                //

            DeleteCriticalSection(&hConnectCriticalSection);
                                    }


            DeleteCriticalSection(&TermCriticalSection);

                                    if
(hLibInstanceTm != NULL)

            FreeLibrary( hLibInstanceTm );
                                                hLibInstanceTm
= NULL;

                                    if
(hLibInstanceDb != NULL)

            FreeLibrary( hLibInstanceDb );
                                                hLibInstanceDb =
NULL;

                                                Sleep(500);
```

```
                                        break;

                                default:
                                        /* nothing */;
                        }
                }
                catch (CBaseErr *e)
                {
                        TCHAR szMsg[256];

                        _sntprintf(szMsg, sizeof(szMsg), "%s error,
code %d: %s",
                                        e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                        WriteMessageToEventLog( szMsg );
                        delete e;
                        TerminateExtension(0);
                        return FALSE;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception.  DLL could not load."));
                        TerminateExtension(0);
                        return FALSE;
                }

                return TRUE;
}


/* FUNCTION: GetExtensionVersion
 *
 * PURPOSE: This function is called by the inet service when the DLL is
first loaded.
 *
 * ARGUMENTS:           HSE_VERSION_INFO        *pVer
                passed in structure in which to place expected version
number.
 *
 * RETURNS:             TRUE            inet service expected return
value.
 */

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
                pVer->dwExtensionVersion =
MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
                lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

                // TODO: why do we need this here instead of in the DLL
attach?         if (Reg.eTxnMon == ENCINA)
                        pCTPCC_ENCINA_post_init();

                return TRUE;
}

/* FUNCTION: TerminateExtension
 *
 * PURPOSE: This function is called by the inet service when the DLL is
about to be unloaded.
 *                                      Release all resources in
anticipation of being unloaded.
 *
 * RETURNS:             TRUE            inet service expected return
value.
 */

BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
                if (pDeliHandles)
                {
                        SetEvent( hDoneEvent );
                        for(DWORD i=0; i<dwNumDeliveryThreads;
i++)
                                        WaitForSingleObject(
pDeliHandles[i], INFINITE );
                }

                TermDeleteAll();
                return TRUE;
}


/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE: This function is the main entry point for the TPCC DLL.
The internet service
 *                                      calls this function passing in
the http string.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
                *pECB           structure pointer to passed in internet
 *
                                        service information.
 *
 * RETURNS:             DWORD           HSE_STATUS_SUCCESS
                                                connection can
be dropped if error
 *
                HSE_STATUS_SUCCESS_AND_KEEP_CONN
                keep connect valid comment sent
 * COMMENTS:            None
 *
 */

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK
*pECB)
{
                int                                     iCmd, FormId,
TermId, iSyncId;
                char                    szBuffer[4096];

                int                                     lpbSize;
                static char     szHeader[] = "200 Ok";
                DWORD                   dwSize = 6;             //
initial value is strlen(szHeader)
                char                    szHeader1[4096];


#ifdef ICECAP
                StartCAP();
#endif

                try
                {
                        //process http query
                        ProcessQueryString(pECB, &iCmd, &FormId,
&TermId, &iSyncId);

                        if (TermId != 0)
                        {
                                if ( TermId < 0 || TermId >=
Term.iNumEntries || Term.pClientData[TermId].iNextFree != -1 )
                                {
                                        // debugging...
                                        char szTmp[128];
                                        wsprintf( szTmp,
"Invalid term ID; TermId = %d", TermId );

                WriteMessageToEventLog( szTmp );

                                        throw new
CWEBCLNT_ERR( ERR_INVALID_TERMID );
                                }

                                //must have a valid syncid here
since termid is valid
                                if (iSyncId !=
Term.pClientData[TermId].iSyncId)
                                        throw new
CWEBCLNT_ERR( ERR_INVALID_SYNC_CONNECTION );

                                //set use time
```

```
		Term.pClientData[TermId].iTickCount = GetTickCount();
		}


		switch(iCmd)
		{
		case 0:
			WelcomeForm(pECB, szBuffer);
			break;
		case 1:
			switch( FormId )
			{
				case
WELCOME_FORM:
				case
MAIN_MENU_FORM:

		break;
				case
NEW_ORDER_FORM:

		ProcessNewOrderForm(pECB, TermId, szBuffer);

		break;
				case
PAYMENT_FORM:

		ProcessPaymentForm(pECB, TermId, szBuffer);

		break;
				case
DELIVERY_FORM:

		ProcessDeliveryForm(pECB, TermId, szBuffer);

		break;
				case
ORDER_STATUS_FORM:

		ProcessOrderStatusForm(pECB, TermId, szBuffer);

		break;
				case
STOCK_LEVEL_FORM:

		ProcessStockLevelForm(pECB, TermId, szBuffer);

		break;
			}
			break;
		case 2:
```

```
				// new-order selected from
menu; display new-order input form	MakeNewOrderForm(TermId,
NULL, INPUT_FORM, szBuffer);
				break;
			case 3:
				// payment selected from
menu; display payment input form
				MakePaymentForm(TermId,
NULL, INPUT_FORM, szBuffer);
				break;
			case 4:
				// delivery selected from menu;
display delivery input form
				MakeDeliveryForm(TermId,
NULL, INPUT_FORM, szBuffer);
				break;
			case 5:
				// order-status selected from
menu; display order-status input form
				MakeOrderStatusForm(TermId,
NULL, INPUT_FORM, szBuffer);
				break;
			case 6:
				// stock-level selected from
menu; display stock-level input form
				MakeStockLevelForm(TermId,
NULL, INPUT_FORM, szBuffer);
				break;
			case 7:
				// ExitCmd
				TermDelete(TermId);
				WelcomeForm(pECB, szBuffer);
				break;
			case 8:
				SubmitCmd(pECB, szBuffer);
				break;
			case 9:
				// menu
				MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
				break;
			case 10:
				// CMD=Clear
				// resets all connections;
should only be used when no other connections are active
				TermDeleteAll();
				TermInit();
				WelcomeForm(pECB, szBuffer);
				break;
			case 11:	// CMD=Stats
				StatsCmd(pECB, szBuffer);
				break;
		}
```

```
		catch (CBaseErr *e)
		{
			ErrorForm( pECB, e->ErrorType(), e-
>ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
			delete e;
		}
		catch (...)
		{
			ErrorForm( pECB, ERR_TYPE_WEBDLL, 0,
TermId, iSyncId, "Error: Unhandled exception in Web Client.", szBuffer
);
		}

#ifdef ICECAP
		StopCAP();
#endif


		lpbSize = strlen(szBuffer);
		wsprintf(szHeader1,
				"Content-Type: text/html\r\n"
				"Content-Length: %d\r\n"
				"Connection: Keep-
Alive\r\n\r\n" , lpbSize);
		strcat( szHeader1, szBuffer );

		(*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize,
(LPDWORD)szHeader1);

		//finish up and keep connection
		pECB->dwHttpStatusCode = 200;
		return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}


void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));

   _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
		{
     ReportEvent(hEventSource, // handle of event source
```

```
        EVENTLOG_ERROR_TYPE,  // event type
        0,                    // event category
        0,                    // event ID
        NULL,                 // current user's SID
        2,                    // strings in lpszStrings
        0,                    // no bytes of raw data
        (LPCTSTR *)lpszStrings,    // array of error strings
        NULL);                // no raw data

    (VOID) DeregisterEventSource(hEventSource);
  }
}


/* FUNCTION: DeliveryWorkerThread
 *
 * PURPOSE: This function processes deferred delivery txns.  There are typically several
 *                                 threads running this routine.  The number of threads is determined by an entry
 *                                 read from the registry.  The thread waits for work by waiting on semaphore.
 *                                 When a delivery txn is posted, the semaphore is released.  After processing
 *                                 the delivery txn, information is logged to record the txn status and execution
 *                                 time.
 */

/*static*/ void DeliveryWorkerThread(void *ptr)
{
        CTPCC_BASE              *pTxn = NULL;

        DELIVERY_TRANSACTION            delivery;
        PDELIVERY_DATA
        pDeliveryData;
        TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

        DWORD
        index;
        HANDLE
        handles[2];

        SYSTEMTIME              trans_end;
        //delivery transaction finished time
        SYSTEMTIME              trans_start;  //delivery transaction start time

        assert(txnDelilog != NULL);

        try
        {
                if (Reg.eDB_Protocol == ODBC)
                        if (Reg.dwConnectDelay > 0)
                        {
                                // Synchronize connect (for VIA)
                                //

                EnterCriticalSection(&hConnectCriticalSection);


                Sleep(Reg.dwConnectDelay);

                                pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,


                szMyComputerName, Reg.szDbName,


                Reg.szSPPrefix, Reg.bCallNoDuplicatesNewOrder );


                LeaveCriticalSection(&hConnectCriticalSection);
                                }
                        }
                        else
                        {
                                if (Reg.eDB_Protocol == DBLIB)
                                pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                        }
                        pDeliveryData = pTxn->BuffAddr_Delivery();
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "Error in Delivery Txn thread.  Could not connect to database.  "
                                "%s. Server=%s, User=%s, Password=%s, Database=%s",
                                e->ErrorText(), Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                        WriteMessageToEventLog( szTmp );
                        delete e;
                        goto ErrorExit;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception caught in DeliveryWorkerThread."));
                }               goto ErrorExit;

                while (TRUE)
                {
                        try
                        {
                                //while delivery thread running, i.e. user has not requested termination
                                while (TRUE)
                                {
                                        // need to wait for multiple objects: program exit or worker semaphore;
                                        handles[0] = hDoneEvent;
                                        handles[1] = hWorkerSemaphore;
                                        index = WaitForMultipleObjects( 2, &handles[0], FALSE, INFINITE );
                                        if (index == WAIT_OBJECT_0)
                                                goto ErrorExit;


                                ZeroMemory(&txnDeliRec, sizeof(txnDeliRec));

                                txnDeliRec.TxnType = TXN_REC_TYPE_TPCC_DELIV_DEF;

                                        // make a local copy of current entry from delivery buffer and increment buffer index

                                EnterCriticalSection(&DelBuffCriticalSection);
                                        delivery = *(pDelBuff+dwDelBuffBusyIndex);

                                dwDelBuffFreeCount++;

                                dwDelBuffBusyIndex++;
                                        if (dwDelBuffBusyIndex == dwDelBuffSize)        // wrap-around if at end of buffer

                                dwDelBuffBusyIndex = 0;


                                LeaveCriticalSection(&DelBuffCriticalSection);

                                        pDeliveryData->w_id = delivery.w_id;
                                        pDeliveryData->o_carrier_id = delivery.o_carrier_id;
```

```
                                        txnDeliRec.w_id
= pDeliveryData->w_id;

        txnDeliRec.o_carrier_id = pDeliveryData->o_carrier_id;

        txnDeliRec.TxnStartT0 = Get64BitTime(&delivery.queue);

                                        GetLocalTime(
&trans_start );
                                        pTxn->Delivery();
                                        GetLocalTime(
&trans_end );

                                        //log txn

        txnDeliRec.TxnStatus = ERR_SUCCESS;
                                        for (int i=0; i<10;
i++)

        txnDeliRec.o_id[i] = pDeliveryData->o_id[i];

        txnDeliRec.DeltaT4 = (int)(Get64BitTime(&trans_end) -
txnDeliRec.TxnStartT0);

        txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                        if (txnDelilog !=
NULL)

        txnDelilog->WriteToLog(&txnDeliRec);
                                }
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "%s Error
(code %d) in Delivery Txn thread. %s",
                                                e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText() );
                        WriteMessageToEventLog(
szTmp );

                        // log the error txn
                        txnDeliRec.TxnStatus = e-
>ErrorType();
                        if (txnDelilog != NULL)
                                txnDelilog-
>WriteToLog(&txnDeliRec);

                        delete e;
                }
                catch (...)
                                // unhandled exception;
shouldn't happen; not much we can do...

        WriteMessageToEventLog(TEXT("Unhandled exception
caught in DeliveryWorkerThread."));
                }
        }

ErrorExit:
        if (Reg.dwConnectDelay > 0)
        {
                // Synchronize disconnect (for VIA)
                //

        EnterCriticalSection(&hConnectCriticalSection);

                Sleep(Reg.dwConnectDelay);

                delete pTxn;


        LeaveCriticalSection(&hConnectCriticalSection);
        }

        _endthread();
}

/* FUNCTION: PostDeliveryInfo
 *
 * PURPOSE: This function enters the delivery txn into the deferred
delivery buffer.
 *
 * RETURNS:             BOOL        FALSE       delivery
information posted successfully
 *
        TRUE        error cannot post delivery info
 */

BOOL PostDeliveryInfo(long w_id, short o_carrier_id)
{
        BOOL bError;

        EnterCriticalSection(&DelBuffCriticalSection);
        if (dwDelBuffFreeCount > 0)
        {
                bError = FALSE;
                (pDelBuff+dwDelBuffFreeIndex)->w_id
                = w_id;
                (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id= o_carrier_id;

        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

                dwDelBuffFreeCount--;
                dwDelBuffFreeIndex++;
                if (dwDelBuffFreeIndex == dwDelBuffSize)
                        dwDelBuffFreeIndex = 0;
        // wrap-around if at end of buffer
        }
        else
                // No free buffers.  Return an error, which
indicates that the delivery buffer is full.
                // Most likely, the number of delivery worker
threads needs to be increased to keep up
                // with the txn rate.
                bError = TRUE;
        LeaveCriticalSection(&DelBuffCriticalSection);

        if (!bError)
                // increment worker semaphore to wake up
a worker thread
                ReleaseSemaphore( hWorkerSemaphore, 1,
NULL );

        return bError;
}

/* FUNCTION: ProcessQueryString
 *
 * PURPOSE: This function extracts the relevent information out of the
http command passed in from
 *                                      the browser.
 *
 * COMMENTS:        If this is the initial connection i.e. client is at
welcome screen then
 *                                              there will not be
a terminal id or current form id.  If this is the case
 *                                              then the pTermid
and pFormid return values are undefined.
 */

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int
*pCmd, int *pFormId, int *pTermId, int *pSyncId)
{
        char *ptr = pECB->lpszQueryString;
        char szBuffer[25];
        int i;

        //allowable client command strings i.e. CMD=command
        static char *szCmds[] =
        {
                "Process", "..NewOrder..", "..Payment..",
"..Delivery..", "..Order-Status..", "..Stock-Level..",
                "..Exit..", "Submit", "Menu", "Clear", "Stats",
""
        };
```

```
		*pCmd		= 0;				// default is the
login screen
		*pTermId = 0;

		// if no params (i.e., empty query string), then return
login screen
		if (strlen(pECB->lpszQueryString) == 0)
			return;

		// parse FORMID, TERMID, and SYNCID
		*pFormId = GetIntKeyValue(&ptr, "FORMID", NO_ERR,
NO_ERR);
		*pTermId = GetIntKeyValue(&ptr, "TERMID", NO_ERR,
NO_ERR);
		*pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR,
NO_ERR);

		// parse CMD
		GetKeyValue(&ptr, "CMD", szBuffer, sizeof(szBuffer),
ERR_COMMAND_UNDEFINED);

		// see which command it matches
		for(i=0; ; i++)
		{
			if (szCmds[i][0] == 0)
				// no more; no match; return
error
				throw new CWEBCLNT_ERR(
ERR_COMMAND_UNDEFINED );
			if ( !strcmp(szCmds[i], szBuffer) )
			{
				*pCmd = i+1;
				break;
			}
		}
}

/* FUNCTION: void WelcomeForm
 *
 */

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer)
{
		char szTmp[1024];

		//welcome to tpc-c html form buffer, this is first form
client sees.
		strcpy( szBuffer,
		"<HTML><HEAD><TITLE>TPC-C Web
Client</TITLE></HEAD><BODY>"

		"<B><BIG>Microsoft TPC-C Web Client (ver
4.20)</BIG></B> <BR> <BR>"
		"<font face=\"Courier New\"><PRE>"

		"Compiled: "__DATE__", "__TIME__" <BR>"

		"Source:   "__FILE__" ("__TIMESTAMP__") <BR>"

		"</PRE></font>"

		"<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"

		"<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"

		"<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"

		"<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"1\">"

		"<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"0\">"

		"<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"0\">"

		"<INPUT TYPE=\"hidden\" NAME=\"VERSION\"
VALUE=\"" WEBCLIENT_VERSION "\">"
					);

		sprintf( szTmp,		"Configuration Settings:
<BR><font face=\"Courier New\" color=\"blue\"><PRE>"

		"Txn Monitor		= <B>%s</B><BR>"

		"Database protocol	= <B>%s</B><BR>"

		"Max Connections	= <B>%d</B><BR>"
"#
of Delivery Threads  = <B>%d</B><BR>"

		"Max Pending Deliveries = <B>%d</B><BR>"
,
szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
		strcat( szBuffer, szTmp);

		if (Reg.eTxnMon == COM)
		{
			sprintf( szTmp,		"COM Single Pool
= <B>%s</B><BR>",		Reg.bCOM_SinglePool ? "YES" :
"NO" );
			strcat( szBuffer, szTmp);
		}
		strcat( szBuffer, "</PRE></font>");

		if (Reg.eTxnMon == None)
			// connection options may be specified when
not using a txn monitor
			sprintf( szTmp,		"Please enter
your database options for this connection:<BR>"

		"<font face=\"Courier New\" color=\"blue\"><PRE>"

		"DB Server    = <INPUT NAME=\"db_server\" SIZE=20
VALUE=\"%s\"><BR>"

		"DB User ID   = <INPUT NAME=\"db_user\" SIZE=20
VALUE=\"%s\"><BR>"

		"DB Password  = <INPUT NAME=\"db_passwd\" SIZE=20
VALUE=\"%s\"><BR>"

		"DB Name      = <INPUT NAME=\"db_name\" SIZE=20
VALUE=\"%s\"><BR>"

		"</PRE></font>"
,
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
		else
			// if using a txn monitor, connection options
are determined from registry; can't
			// set per user.  show options fyi
			sprintf( szTmp,		"Database
options which will be used by the transaction monitor:<BR>"

		"<font face=\"Courier New\" color=\"blue\"><PRE>"

		"DB Server		= <B>%s</B><BR>"

		"DB User ID		= <B>%s</B><BR>"

		"DB Password		= <B>%s</B><BR>"

		"DB Name		= <B>%s</B><BR>"

		"</PRE></font>"
,
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
		strcat( szBuffer, szTmp);
```

```
		sprintf( szTmp,		"Please enter your Warehouse
and District for this session:<BR>"

		"<font face=\"Courier New\" color=\"blue\"><PRE>" );
		strcat( szBuffer, szTmp);
		strcat( szBuffer,		"Warehouse ID = <INPUT
NAME=\"w_id\" SIZE=6><BR>"

		"District ID  = <INPUT NAME=\"d_id\" SIZE=2><BR>"

		"</PRE></font><HR>"

		"<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Submit\">"

		"</FORM></BODY></HTML>");
}

/* FUNCTION: SubmitCmd
 *
 * PURPOSE: This function allocated a new terminal id in the Term
structure array.
 *
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
		int			iNewTerm;
		char		*ptr = pECB->lpszQueryString;

		char		szVersion[32]		= { 0 };
		char		szServer[32] = { 0 };
		char		szUser[32]		= "sa";
		char		szPassword[32]		= { 0 };
		char		szDatabase[32]		= "tpcc";

		// validate version field; the version field ensures that the
RTE is synchronized with the web client
		GetKeyValue(&ptr, "VERSION", szVersion,
sizeof(szVersion), ERR_VERSION_MISMATCH);
		if ( strcmp( szVersion, WEBCLIENT_VERSION ) )
			throw new CWEBCLNT_ERR(
ERR_VERSION_MISMATCH );

		if (Reg.eTxnMon == None)
		{
			// parse Server name
			GetKeyValue(&ptr, "db_server", szServer,
sizeof(szServer), ERR_NO_SERVER_SPECIFIED);
			// parse User name
			GetKeyValue(&ptr, "db_user", szUser,
sizeof(szUser), NO_ERR);
			// parse Password
			GetKeyValue(&ptr, "db_passwd",
szPassword, sizeof(szPassword), NO_ERR);
			// parse Database name
			GetKeyValue(&ptr, "db_name", szDatabase,
sizeof(szDatabase), NO_ERR);
		}

		// parse warehouse ID
		int w_id = GetIntKeyValue(&ptr, "w_id",
ERR_HTML_ILL_FORMED, ERR_W_ID_INVALID);
		if ( w_id < 1 )
			throw new CWEBCLNT_ERR(
ERR_W_ID_INVALID );

		// parse district ID
		int d_id = GetIntKeyValue(&ptr, "d_id",
ERR_HTML_ILL_FORMED, ERR_D_ID_INVALID);
		if ( d_id < 1 || d_id > 10 )
			throw new CWEBCLNT_ERR(
ERR_D_ID_INVALID );


		iNewTerm = TermAdd();

		Term.pClientData[iNewTerm].w_id = w_id;
		Term.pClientData[iNewTerm].d_id = d_id;

		try
		{
			if (Reg.eTxnMon == TUXEDO)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_TUXEDO_new();
			else if (Reg.eTxnMon == ENCINA)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_ENCINA_new();
			else if (Reg.eTxnMon == COM)

		Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new(
Reg.bCOM_SinglePool );
			else if (Reg.eDB_Protocol == ODBC)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_ODBC_new(	szServer, szUser, szPassword,
szMyComputerName,



		szDatabase, Reg.szSPPrefix,
		Reg.bCallNoDuplicatesNewOrder );

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_DBLIB_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase );
		}
		catch (...)
		{
			TermDelete(iNewTerm);
			throw;			// pass exception
upward
		}

		MakeMainMenuForm(iNewTerm,
Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

/* FUNCTION: StatsCmd
 *
 * PURPOSE: This function returns to the browser the total number of
active terminal ids.
 *			This routine is for
development/debugging purposes.
 *
 */

void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
		int i;
		int		iTotal;

		EnterCriticalSection(&TermCriticalSection);

		iTotal = 0;
		for(i=0; i<Term.iNumEntries; i++)
		{
			if (Term.pClientData[i].iNextFree == -1)
				iTotal++;
		}

		LeaveCriticalSection(&TermCriticalSection);

		wsprintf( szBuffer,

		"<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"
			"<BODY><B><BIG> Total
Active Connections: %d </BIG></B><BR></BODY></HTML>"
			, iTotal );
}
```

```
char *CWEBCLNT_ERR::ErrorText()
{
        static SERRORMSG errorMsgs[] =
        {
                {       ERR_COMMAND_UNDEFINED,

        "Command undefined."

                                        },
                {       ERR_D_ID_INVALID,

        "Invalid District ID Must be 1 to 10."

        },
                {
        ERR_DELIVERY_CARRIER_ID_RANGE,
        "Delivery Carrier ID out of range must be 1 - 10."
                                        },
                {
        ERR_DELIVERY_CARRIER_INVALID,
        "Delivery Carrier ID invalid must be numeric 1 - 10."
                                },
                {
        ERR_DELIVERY_MISSING_OCD_KEY,
        "Delivery missing Carrier ID key \"OCD*\"."

        },
                {
        ERR_DELIVERY_THREAD_FAILED,
                "Could not start delivery worker thread."

        },
                {       ERR_GETPROCADDR_FAILED,

        "Could not map proc in DLL.  GetProcAddr error.  DLL="
                                        },
                {       ERR_HTML_ILL_FORMED,
                                "Required key
field is missing from HTML string."
                },
                {
        ERR_INVALID_SYNC_CONNECTION,
        "Invalid Terminal Sync ID."

                                },
                {       ERR_INVALID_TERMID,

        "Invalid Terminal ID."

                                        },
                {       ERR_LOADDLL_FAILED,

        "Load of DLL failed.  DLL="

                                {               },
        ERR_MAX_CONNECTIONS_EXCEEDED,
        "No connections available.  Max Connections is probably
too low."   },
                {
        ERR_MISSING_REGISTRY_ENTRIES,
        "Required registry entries are missing.  Rerun INSTALL to
correct."       },
                {
        ERR_NEWORDER_CUSTOMER_INVALID,
        "New Order customer id invalid data type, range = 1 to
3000."          },
                {
        ERR_NEWORDER_CUSTOMER_KEY,
                "New Order missing Customer key \"CID*\"."

                },
                {
        ERR_NEWORDER_DISTRICT_INVALID,
        "New Order District ID Invalid range 1 - 10."
                                                },
                {
        ERR_NEWORDER_FORM_MISSING_DID,
        "New Order missing District key \"DID*\"."

        },
                {
        ERR_NEWORDER_ITEMID_INVALID,
        "New Order Item Id is wrong data type, must be
numeric."                       },
                {
        ERR_NEWORDER_ITEMID_RANGE,
                "New Order Item Id is out of range. Range =
1 to 999999."                           },
                {
        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        "New Order Item_Id field entered without a corresponding
Supp_W."        },
                {
        ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key \"IID*\"."

        },
                {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key \"Qty##*\"."

                },
                {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
                "New Order missing Supp_W key
\"SP##*\"."
                },
        ERR_NEWORDER_NOITEMS_ENTERED,
        "New Order No order lines entered."

                },
                {
        ERR_NEWORDER_QTY_INVALID,
                "New Order Qty invalid must be numeric
range 1 - 99."                                          },
                {
        ERR_NEWORDER_QTY_RANGE,
                        "New Order Qty is out of
range. Range = 1 to 99."
        },
                {
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                "New Order Qty field entered without a
corresponding Supp_W."          },
                {
        ERR_NEWORDER_SUPPW_INVALID,
                "New Order Supp_W invalid data type must
be numeric."                                    },
                {       ERR_NO_SERVER_SPECIFIED,
                                "No Server name
specified."

        },
                {
        ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name may be
entered, not both."     },
                {
        ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must be numeric
1 - 3000."      },
                {
        ERR_ORDERSTATUS_CLT_RANGE,
                "Order Status Customer last name longer
than 16 characters."            },
                {
        ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be numeric 1 -
10."            },
                {
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last Name must be
entered."              },
```

```
                {
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key \"CID*\"."
                                                },
                {
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name key \"CLT*\"."
                                                },
                {
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key \"DID*\"."
                                                },
                {       ERR_PAYMENT_CDI_INVALID,
                                                "Payment
Customer district invalid must be numeric."
                },
                {       ERR_PAYMENT_CID_AND_CLT,
                                                "Payment Only
Customer ID or Last Name may be entered, not both."             },
                {
        ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."
                                                },
                {       ERR_PAYMENT_CWI_INVALID,
                                                "Payment
Customer Warehouse invalid, must be numeric."
                },
                {
        ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1 - 10."
                                                },
                {       ERR_PAYMENT_HAM_INVALID,
                                                "Payment Amount
invalid data type must be numeric."
                },
                {       ERR_PAYMENT_HAM_RANGE,

        "Payment Amount out of range, 0 - 9999.99."

        },
                {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."
                                                },
                {
        ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key \"CDI*\"."
                                                },
                {
        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name must be
entered."                       },
```

```
                {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \"CID*\"."

        },
                {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key \"CLT*\"."
                                                },
                {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key \"CWI*\"."
                                                },
                {
        ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \"DID*\"."

        },
                {
        ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key \"HAM*\"."

                },
                {
        ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
        "Stock Level; missing Threshold key \"TT*\"."
                                                },
                {
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in the range = 1 -
99."              },
                {
        ERR_STOCKLEVEL_THRESHOLD_RANGE,
                "Stock Level Threshold out of range, range
must be 1 - 99."                                },
                {       ERR_VERSION_MISMATCH,

        "Invalid version field.  RTE and Web Client are probably
out of sync." },
                {       ERR_W_ID_INVALID,

        "Invalid Warehouse ID."

                                                },
                {       0,

                        ""



                                                }
        };
```

```
        int i = 0;
        char szTmp[256];
        while (TRUE)
        {
                if (errorMsgs[i].szMsg[0] == 0)
                {
                        strcpy( szTmp, "Unknown error
number." );
                        break;
                }
                if (m_Error == errorMsgs[i].iError)
                {
                        strcpy( szTmp,
errorMsgs[i].szMsg );
                        break;
                }
                i++;
        }

        if (m_szTextDetail)
                strcat( szTmp, m_szTextDetail );
        if (m_SystemErr)
                wsprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

        m_szErrorText = new char[strlen(szTmp)+1];
        strcpy( m_szErrorText, szTmp );
        return m_szErrorText;

}

/* FUNCTION: GetKeyValue
 *
 * PURPOSE: This function parses a http formatted string for specific
key values.
 *
 * ARGUMENTS:           char
        *pQueryString           http string from client browser
 *                                      char
                *pKey                                   key
value to look for
 *                                      char
                *pValue
        character array into which to place key's value
 *                                      int
                        iMax
        maximum length of key value array.
 *                                      WEBERROR
        err
        error value to throw
 *
 * RETURNS:             nothing.
 *
 * ERROR:               if (the pKey value is not found) then
```

```
 *                                                                    if
(err == 0)
 *
                return (empty string)
 *                                                                    else
 *
                throw CWEBCLNT_ERR(err)
 *
 * COMMENTS:              http keys are formatted either KEY=value&
or KEY=value\0. This DLL formats
 *                                                          TPC-C input fields
in such a manner that the keys can be extracted in the
 *                                                          above manner.
 */

void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int
iMax, WEBERROR err)
{
                char *ptr;

                if ( !(ptr=strstr(*pQueryString, pKey)) )
                                goto ErrorExit;
                ptr += strlen(pKey);
                if ( *ptr != '=' )
                                goto ErrorExit;
                ptr++;

                iMax--;  // one position is for terminating null
                while( *ptr && *ptr != '&' && iMax)
                {
                                *pValue++ = *ptr++;
                                iMax--;
                }
                *pValue = 0;  // terminating null

                *pQueryString = ptr;
                return;

ErrorExit:
                if (err != NO_ERR)
                                throw new CWEBCLNT_ERR( err );
                *pValue = 0;  // return empty result string
}

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE: This function parses a http formatted string for a specific
key value.
 *
 * ARGUMENTS:             char
                *pQueryString              http string from client browser
 *                                                          char
                                *pKey                                   key
value to look for
                                                            WEBERROR
                NoKeyErr                   error value to throw if key not
found
 *                                                          WEBERROR
                NotIntErr                  error value to throw if value
not numeric
 *
 * RETURNS:               integer
 *
 * ERROR:                 if (the pKey value is not found) then
 *                                                                    if
(NoKeyErr != NO_ERR)
 *
                throw CWEBCLNT_ERR(err)
 *                                                                    else
 *
                return 0
 *                                                          else if (non-
numeric char found) then
 *                                                                    if
(NotIntErr != NO_ERR) then
 *
                throw CWEBCLNT_ERR(err)
 *                                                                    else
 *
                return 0
 *
 * COMMENTS:              http keys are formatted either KEY=value&
or KEY=value\0. This DLL formats
 *                                                          TPC-C input fields
in such a manner that the keys can be extracted in the
 *                                                          above manner.
 */

int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR
NoKeyErr, WEBERROR NotIntErr)
{
                char *ptr0;
                char *ptr;

                if ( !(ptr=strstr(*pQueryString, pKey)) )
                                goto ErrorNoKey;
                ptr += strlen(pKey);
                if ( *ptr != '=' )
                                goto ErrorNoKey;
                ptr++;

                ptr0 = ptr;                // remember starting point
                // scan string until a terminator (null or &) or a non-digit
                while( *ptr && *ptr != '&' && isdigit(*ptr) )
                                ptr++;
                // make sure we stopped scanning for the right reason
                if ((ptr0 == ptr) || (*ptr && *ptr != '&'))
                {
                                if (NotIntErr != NO_ERR)
                                                throw new CWEBCLNT_ERR(
NoKeyErr );
                                return 0;
                }

                *pQueryString = ptr;
                return atoi(ptr0);

ErrorNoKey:
                if (NoKeyErr != NO_ERR)
                                throw new CWEBCLNT_ERR( NoKeyErr );
                return 0;
}

/* FUNCTION: TermInit
 *
 * PURPOSE: This function initializes the client terminal structure; it is
called when the TPCC.DLL
 *                                            is first loaded by the inet
service.
 *
 */

void TermInit(void)
{
                EnterCriticalSection(&TermCriticalSection);

                Term.iMasterSyncId          = 1;
                Term.iNumEntries            = Reg.dwMaxConnections+1;

                Term.pClientData            = NULL;
                Term.pClientData            =
(PCLIENTDATA)malloc(Term.iNumEntries * sizeof(CLIENTDATA));
                if (Term.pClientData == NULL)
                {
                                LeaveCriticalSection(&TermCriticalSection);
                                throw new CWEBCLNT_ERR(
ERR_MEM_ALLOC_FAILED );
                }

                ZeroMemory( Term.pClientData, Term.iNumEntries *
sizeof(CLIENTDATA) );

                Term.iFreeList                                  =
Term.iNumEntries-1;
                // build free list
                // note: Term.pClientData[0].iNextFree gets set to -1,
which marks it as "in use".
```

```
		//	This is intentional, as the zero entry is used as an
anchor and never
		//				allocated as an actual
terminal.
		for(int i=0; i<Term.iNumEntries; i++)
			Term.pClientData[i].iNextFree = i-1;

		LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll
 *
 * PURPOSE: This function frees allocated resources associated with the
terminal structure.
 *
 * ARGUMENTS:		none
 *
 * RETURNS:		None
 *
 * COMMENTS:		This function is called only when the inet
service unloads the TPCC.DLL
 *
 */

void TermDeleteAll(void)
{
		EnterCriticalSection(&TermCriticalSection);

		for(int i=1; i<Term.iNumEntries; i++)
		{
			if (Term.pClientData[i].iNextFree == -1)
				delete
Term.pClientData[i].pTxn;
		}

		Term.iFreeList				= 0;
		Term.iNumEntries		= 0;
		if ( Term.pClientData )
			free(Term.pClientData);
		Term.pClientData		= NULL;

		LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermAdd
 *
 * PURPOSE: This function assigns a terminal id which is used to
identify a client browser.
 *
 * RETURNS:		int				assigned terminal
id
 *
 */
int TermAdd(void)
{
		DWORD	i;
		int				iNewTerm, iTickCount;

		if (Term.iNumEntries == 0)
			return -1;

		EnterCriticalSection(&TermCriticalSection);
		if (Term.iFreeList != 0)
		{		// position is available
			iNewTerm = Term.iFreeList;
			Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;
			Term.pClientData[iNewTerm].iNextFree = -
1;  // indicates this position is in use
		}
		else
		{		// no open slots, so find the slot that hasn't
been used in the longest time and reuse it
			for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
			{
				if (iTickCount >
Term.pClientData[i].iTickCount)
				{
					iTickCount =
Term.pClientData[i].iTickCount;
					iNewTerm = i;
				}
			}
			// if oldest term is less than one minute old,
it probably means that more connections
			// are being attempted than were specified
as "Max Connections" at install.  In this case,
			// do not bump existing connection; instead,
return error to requestor.
			if ((GetTickCount() - iTickCount) < 60000)
			{

		LeaveCriticalSection(&TermCriticalSection);
				throw new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
			}
		}

		Term.pClientData[iNewTerm].iTickCount =
GetTickCount();
		Term.pClientData[iNewTerm].iSyncId =
Term.iMasterSyncId++;
		Term.pClientData[iNewTerm].pTxn = NULL;
		return iNewTerm;
		LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDelete
 *
 * PURPOSE: This function makes a terminal entry in the Term array
available for reuse.
 *
 * ARGUMENTS:		int
id
		Terminal id of client exiting
 *
 */

void TermDelete(int id)
{
		if ( id > 0 && id < Term.iNumEntries )
		{
			delete Term.pClientData[id].pTxn;

			// put onto free list
			EnterCriticalSection(&TermCriticalSection);

			Term.pClientData[id].iNextFree =
Term.iFreeList;
			Term.iFreeList = id;

			LeaveCriticalSection(&TermCriticalSection);
		}
}


/* FUNCTION: MakeErrorForm
 */

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
		wsprintf(szBuffer,
			"<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
			"<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
```

```
                        "<BOLD>An Error
Occurred</BOLD><BR><BR>"
                        "%s"
                        "<BR><BR><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
                        "</FORM></BODY></HTML>"
                        , iType, iErrorNum, MAIN_MENU_FORM,
iTermId, iSyncId, szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */

void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm)
{
        wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Main
Menu</TITLE></HEAD><BODY>"
                        "Select Desired Transaction.<BR><HR>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"    "</FORM></BODY></HTML>"
                        , MAIN_MENU_FORM, iTermId, iSyncId);
}

/* FUNCTION: MakeStockLevelForm
 *
 * PURPOSE: This function constructs the Stock Level HTML page.
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */

void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Stock-Level<BR>"
                        "Warehouse: %6.6d   District: %2.2d<BR>
<BR>",
                        STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
                        Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                                "Stock Level Threshold:
<INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
                                "low stock:    </font><BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"   "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                                "</FORM></HTML>" );
        }
        else
        {
                wsprintf(szForm+c,
                                "Stock Level Threshold:
%2.2d<BR> <BR>"
                                "low stock: %3.3d</font>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                                "</FORM></HTML>"
                                , pStockLevelData->threshold,
pStockLevelData->low_stock);
        }
}

/* FUNCTION: MakeNewOrderForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm)
{
        int                     i, c;
        BOOL            bValid;
        static          char szBR[] = " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>";

        if (!bInput)
                assert( pNewOrderData->exec_status_code
== eOK || pNewOrderData->exec_status_code == eInvalidItem );
```

```
	bValid = (bInput || (pNewOrderData->exec_status_code
== eOK));

	c = wsprintf(szForm,
			"<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD><BODY>"
			"<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
			"<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
			"<PRE><font face=\"Courier\">
New Order<BR>"
			, bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

	if ( bInput )
	{
			c += wsprintf(szForm+c, "Warehouse:
%6.6d   ", Term.pClientData[iTermId].w_id );

			strcpy( szForm+c,
				"District: <INPUT
NAME=\"DID*\" SIZE=1>		Date:<BR>"
				"Customer:  <INPUT
NAME=\"CID*\" SIZE=4>   Name:		Credit:
%Disc:<BR>"
				"Order Number:
Number of Lines:	W_tax:	D_tax:<BR> <BR>"
				" Supp_W  Item_Id  Item
Name		Qty  Stock  B/G  Price    Amount<BR>"
				"  <INPUT NAME=\"SP00*\"
SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP01*\"
SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP02*\"
SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP03*\"
SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP04*\"
SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP05*\"
SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP06*\"
SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP07*\"
SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP08*\"
SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP09*\"
SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP10*\"
SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP11*\"
SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP12*\"
SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP13*\"
SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
				"  <INPUT NAME=\"SP14*\"
SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
				"Execution Status:
Total:<BR>"
				"</font></PRE><HR>"
				"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
				"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
				"</FORM></HTML>"
				);
	}
	else
	{
			c += wsprintf(szForm+c, "Warehouse:
%6.6d   District: %2.2d		Date: ",
				pNewOrderData->w_id,
				pNewOrderData->d_id);

			if ( bValid )
			{
				c += wsprintf(szForm+c,
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
					pNewOrderData-
>o_entry_d.day,
					pNewOrderData-
>o_entry_d.month,
					pNewOrderData-
>o_entry_d.year,
					pNewOrderData-
>o_entry_d.hour,
					pNewOrderData-
>o_entry_d.minute,
					pNewOrderData-
>o_entry_d.second);
			}

			c += wsprintf(szForm+c, "<BR>Customer:
%4.4d   Name: %-16s   Credit: %-2s   ",
				pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);

			if ( bValid )
			{
				c += sprintf(szForm+c,

	"%%Disc: %5.2f		<BR>"

	"Order Number: %8.8d  Number of Lines: %2.2d
W_tax: %5.2f   D_tax: %5.2f  <BR> <BR>"

	" Supp_W  Item_Id  Item Name		Qty  Stock
B/G  Price    Amount<BR>",

	100.0*pNewOrderData->c_discount,
					pNewOrderData-
>o_id,
					pNewOrderData-
>o_ol_cnt,
					100.0 *
pNewOrderData->w_tax,
					100.0 *
pNewOrderData->d_tax);

				for(i=0; i<pNewOrderData-
>o_ol_cnt; i++)
				{
					c +=
sprintf(szForm+c, "%6.6d  %6.6d  %-24s  %2.2d  %3.3d  %1.1s
$%6.2f  $%7.2f  <BR>",

	pNewOrderData->OL[i].ol_supply_w_id,

	pNewOrderData->OL[i].ol_i_id,

	pNewOrderData->OL[i].ol_i_name,
```

```
                pNewOrderData->OL[i].ol_quantity,

                pNewOrderData->OL[i].ol_stock,

                pNewOrderData->OL[i].ol_brand_generic,

                pNewOrderData->OL[i].ol_i_price,

                pNewOrderData->OL[i].ol_amount );
                                }
                        }
                        else
                        {
                                c += wsprintf(szForm+c,
                                        "%Disc:<BR>"
                                        "Order Number:
%8.8d  Number of Lines:          W_tax:        D_tax:<BR> <BR>"
                                        " Supp_W
Item_Id  Item Name               Qty  Stock  B/G  Price    Amount<BR>"
                                        ,
pNewOrderData->o_id);

                                i = 0;
                        }

                        strncpy( szForm+c, szBR, (15-i)*5 );
                        c += (15-i)*5;

                        if ( bValid )
                                c += sprintf(szForm+c,
"Execution Status: Transaction commited.                 Total:  $%8.2f
",
                                        pNewOrderData-
>total_amount);
                        else
                                c += wsprintf(szForm+c,
"Execution Status: Item number is not valid.             Total:");

                        strcpy(szForm+c,
                                " <BR></font></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"    "</FORM></HTML>"
                                );
                }
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData,
BOOL bInput, char *szForm)
{
                int c;

                c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Payment<BR>"
                        "Date: "
                        , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

                if ( !bInput )
                {
                        c += wsprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
                                pPaymentData->h_date.day,
                                pPaymentData-
>h_date.month,
                                pPaymentData->h_date.year,
                                pPaymentData->h_date.hour,
                                pPaymentData-
>h_date.minute,
                                pPaymentData-
>h_date.second);
                }
                if ( bInput )
                {
                        c += wsprintf(szForm+c,
                                "<BR> <BR>Warehouse:
%6.6d"
                                "                District:
<INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR> <BR>"
                                "Customer: <INPUT
NAME=\"CID*\" SIZE=4>"
                                "Cust-Warehouse: <INPUT
NAME=\"CWI*\" SIZE=4>  "
                                "Cust-District: <INPUT
NAME=\"CDI*\" SIZE=1><BR>"
                                "Name:
<INPUT NAME=\"CLT*\" SIZE=16>          Since:<BR>"
                                "
Credit:<BR>"
                                "
Disc:<BR>"
                                "
Phone:<BR> <BR>"
                                "Amount Paid:          $<INPUT
NAME=\"HAM*\" SIZE=7>      New Cust-Balance:<BR>"
                                "Credit Limit:<BR> <BR>Cust-
Data: <BR> <BR> <BR> <BR> <BR></font></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"

                        "</BODY></FORM></HTML>"
                                ,
Term.pClientData[iTermId].w_id);
                }
                else
                {
                        c += wsprintf(szForm+c,
                                "<BR> <BR>Warehouse:
%6.6d                District: %2.2d<BR>"
                                "%-20s                %-
20s<BR>"
                                "%-20s                %-
20s<BR>"
                                "%-20s %-2s %5.5s-%4.4s
%-20s %-2s %5.5s-%4.4s<BR> <BR>"
                                "Customer: %4.4d  Cust-
Warehouse: %6.6d  Cust-District: %2.2d<BR>"
                                "Name:   %-16s %-2s %-16s
Since:  %2.2d-%2.2d-%4.4d<BR>"
                                "       %-20s
Credit: %-2s<BR>"
```

```
                                        ,
Term.pClientData[iTermId].w_id, pPaymentData->d_id
                                        , pPaymentData->w_street_1,
pPaymentData->d_street_1
                                        , pPaymentData->w_street_2,
pPaymentData->d_street_2
                                        , pPaymentData->w_city,
pPaymentData->w_state, pPaymentData->w_zip, pPaymentData-
>w_zip+5
                                        , pPaymentData->d_city,
pPaymentData->d_state, pPaymentData->d_zip, pPaymentData-
>d_zip+5
                                        , pPaymentData->c_id,
pPaymentData->c_w_id,   pPaymentData->c_d_id
                                        , pPaymentData->c_first,
pPaymentData->c_middle, pPaymentData->c_last
                                        , pPaymentData->c_since.day,
pPaymentData->c_since.month,            pPaymentData->c_since.year
                                        , pPaymentData->c_street_1,
pPaymentData->c_credit
                                        );

        c += sprintf(szForm+c,
                                        "       %-20s
%%Disc:  %5.2f<BR>",
                                        pPaymentData->c_street_2,
100.0*pPaymentData->c_discount);

                        c += wsprintf(szForm+c,
                                        "       %-20s %-2s %5.5s-
%4.4s      Phone:  %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                                        pPaymentData->c_city,
pPaymentData->c_state, pPaymentData->c_zip, pPaymentData-
>c_zip+5,
                                        pPaymentData->c_phone,
pPaymentData->c_phone+6, pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );

                        c += sprintf(szForm+c,
                                        "Amount Paid:          $%7.2f
New Cust-Balance: $%14.2f<BR>"
                                        "Credit Limit:   $%13.2f<BR>
<BR>"
                                        , pPaymentData->h_amount,
pPaymentData->c_balance
                                        , pPaymentData->c_credit_lim
                                        );

                        if ( pPaymentData->c_credit[0] == 'B' &&
pPaymentData->c_credit[1] == 'C' )
                                        c += wsprintf(szForm+c,
                "Cust-Data: %-50.50s<BR>        %-50.50s<BR>
%-50.50s<BR>        %-50.50s<BR>",
                pPaymentData->c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData->c_data+150 );
                        else
                                        strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");

                        strcat(szForm,          "
<BR></font></PRE><HR>"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>");
                }

}

/* FUNCTION: MakeOrderStatusForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                be freed except
when the client terminal id is no longer needed.
 */

void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm)
{
        int                     i, c;
        static char szBR[] = " <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Order-Status<BR>"
                        "Warehouse: %6.6d   ",
                        ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                        "District: <INPUT
NAME=\"DID*\" SIZE=1><BR>"
                        "Customer: <INPUT
NAME=\"CID*\" SIZE=4>   Name:           <INPUT
NAME=\"CLT*\" SIZE=23><BR>"
                        "Cust-Balance:<BR> <BR>"
                        "Order-Number:          Entry-
Date:                   Carrier-Number:<BR>"
                        "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
                        "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"

                "</BODY></FORM></HTML>" );
                }
                else
                {
                        c += wsprintf(szForm+c,
                                "District: %2.2d<BR>"
                                "Customer: %4.4d   Name: %-
16s %-2s %-16s<BR>",
                                pOrderStatusData->d_id,
pOrderStatusData->c_id,
                                pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData->c_last);

                        c += sprintf(szForm+c, "Cust-Balance:
$%9.2f<BR> <BR>",
                                pOrderStatusData-
>c_balance);
```

```
                        c += wsprintf(szForm+c,
                                "Order-Number: %8.8d   Entry-
Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d   Carrier-Number:
%2.2d<BR>"
                                "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR>",
                                pOrderStatusData->o_id,
                                pOrderStatusData-
>o_entry_d.day,
                                pOrderStatusData-
>o_entry_d.month,
                                pOrderStatusData-
>o_entry_d.year,
                                pOrderStatusData-
>o_entry_d.hour,
                                pOrderStatusData-
>o_entry_d.minute,
                                pOrderStatusData-
>o_entry_d.second,
                                pOrderStatusData-
>o_carrier_id);

                        for(i=0; i< pOrderStatusData->o_ol_cnt;
i++)
                        {
                                c += sprintf(szForm+c, "
%6.6d     %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-%4.4d<BR>",

                pOrderStatusData->OL[i].ol_supply_w_id,

                pOrderStatusData->OL[i].ol_i_id,

                pOrderStatusData->OL[i].ol_quantity,

                pOrderStatusData->OL[i].ol_amount,

                pOrderStatusData->OL[i].ol_delivery_d.day,

                pOrderStatusData->OL[i].ol_delivery_d.month,

                pOrderStatusData->OL[i].ol_delivery_d.year);
                        }

                        strncpy( szForm+c, szBR, (15-i)*5 );
                        c += (15-i)*5;

                        strcpy(szForm+c,
                                "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>" );
                }
}

/* FUNCTION: MakeDeliveryForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">
Delivery<BR>"
                "Warehouse: %6.6d<BR> <BR>",
                (!bInput && (pDeliveryData-
>exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
                DELIVERY_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy( szForm+c,
                        "Carrier Number: <INPUT
NAME=\"OCD*\" SIZE=1><BR> <BR>"
                        "Execution Status: <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE><HR>"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"

                "</BODY></FORM></HTML>" );
                }
                else
                {
                        wsprintf( szForm+c,
                                "Carrier Number: %2.2d<BR>
<BR>"
                                "Execution Status: %s <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE>"
                                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>"

                                , pDeliveryData->o_carrier_id,
                                (pDeliveryData-
>exec_status_code == eOK) ? "Delivery has been queued." : "Delivery
Post Failed     "
                                );
                }
}

/* FUNCTION: ProcessNewOrderForm
 *
 * PURPOSE: This function gets and validates the input data from the
new order form
 *                                      filling in the required input
variables. it then calls the SQLNewOrder
 *                                      transaction, constructs the
output form and writes it back to client
 *                                      browser.
 */
```

```
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
        *pECB, int iTermId, char *szBuffer)
{
        PNEW_ORDER_DATA                 pNewOrder;

        pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();

        ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
        pNewOrder->w_id = Term.pClientData[iTermId].w_id;
        GetNewOrderData(pECB->lpszQueryString, pNewOrder);

        Term.pClientData[iTermId].pTxn->NewOrder();

        pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();
        MakeNewOrderForm(iTermId, pNewOrder,
OUTPUT_FORM, szBuffer );
}

/* FUNCTION: void ProcessPaymentForm
 *
 * PURPOSE: This function gets and validates the input data from the
payment form
 *                                 filling in the required input
variables. It then calls the SQLPayment
 *                                 transaction, constructs the
output form and writes it back to client
 *                                 browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
        *pECB       passed in structure pointer from inetsrv.
 *                                                 int

        iTermId     client browser terminal id
 *
 */

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
        PPAYMENT_DATA           pPayment;

        pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
        ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
        pPayment->w_id = Term.pClientData[iTermId].w_id;
        GetPaymentData(pECB->lpszQueryString, pPayment);

        Term.pClientData[iTermId].pTxn->Payment();

        pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
        MakePaymentForm(iTermId, pPayment, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
 *
 * PURPOSE: This function gets and validates the input data from the
Order Status
 *                                 form filling in the required
input variables. It then calls the
 *                                 SQLOrderStatus transaction,
constructs the output form and writes it
 *                                 back to client browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
        *pECB       passed in structure pointer from inetsrv.
 *                                                 int

        iTermId     client browser terminal id
 *
 */

void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
        PORDER_STATUS_DATA      pOrderStatus;

        pOrderStatus = Term.pClientData[iTermId].pTxn-
>BuffAddr_OrderStatus();
        ZeroMemory(pOrderStatus,
sizeof(ORDER_STATUS_DATA));
        pOrderStatus->w_id = Term.pClientData[iTermId].w_id;
        GetOrderStatusData(pECB->lpszQueryString,
pOrderStatus);

        Term.pClientData[iTermId].pTxn->OrderStatus();

        pOrderStatus = Term.pClientData[iTermId].pTxn-
>BuffAddr_OrderStatus();
        MakeOrderStatusForm(iTermId, pOrderStatus,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessDeliveryForm
 *
 * PURPOSE: This function gets and validates the input data from the
delivery form
 *                                 filling in the required input
variables. It then calls the PostDeliveryInfo
 *                                 Api, The client is then informed
that the transaction has been posted.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
 *      *pECB       passed in structure pointer from inetsrv.
 *                                                 int

        iTermId     client browser terminal id
 *
 */

void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
        char            *ptr = pECB->lpszQueryString;

        PDELIVERY_DATA          pDelivery;

        pDelivery = Term.pClientData[iTermId].pTxn-
>BuffAddr_Delivery();
        ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
        pDelivery->w_id = Term.pClientData[iTermId].w_id;

        pDelivery->o_carrier_id     = GetIntKeyValue(&ptr,
"OCD*", ERR_DELIVERY_MISSING_OCD_KEY,
ERR_DELIVERY_CARRIER_INVALID);
        if ( pDelivery->o_carrier_id > 10 || pDelivery-
>o_carrier_id < 1 )
                throw new CWEBCLNT_ERR(
ERR_DELIVERY_CARRIER_ID_RANGE );

        if (dwNumDeliveryThreads)
        {
                //post delivery info
                if ( PostDeliveryInfo(pDelivery->w_id,
pDelivery->o_carrier_id) )
                        pDelivery->exec_status_code
= eDeliveryFailed;
                else
                        pDelivery->exec_status_code
= eOK;
        }
        else // delivery is done synchronously if no delivery
threads configured
                Term.pClientData[iTermId].pTxn-
>Delivery();

        pDelivery = Term.pClientData[iTermId].pTxn-
>BuffAddr_Delivery();
        MakeDeliveryForm(iTermId, pDelivery, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
 *
 * PURPOSE: This function gets and validates the input data from the
Stock Level
```

```
 *                                      form filling in the required
input variables. It then calls the
 *                                      SQLStockLevel transaction,
constructs the output form and writes it
 *                                      back to client browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
            *pECB       passed in structure pointer from inetsrv.
 *                                              int

            iTermId     client browser terminal id
 *
 */

void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
            char                    *ptr = pECB->lpszQueryString;

            PSTOCK_LEVEL_DATA       pStockLevel;

            pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
            ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

            pStockLevel->w_id = Term.pClientData[iTermId].w_id;
            pStockLevel->d_id = Term.pClientData[iTermId].d_id;

            pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);
            if ( pStockLevel->threshold >= 100 || pStockLevel-
>threshold < 0 )
                        throw new CWEBCLNT_ERR(
ERR_STOCKLEVEL_THRESHOLD_RANGE );

            Term.pClientData[iTermId].pTxn->StockLevel();

            pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
            MakeStockLevelForm(iTermId, pStockLevel,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *
 * PURPOSE: This function extracts and validates the new order form
data from an http command string.
 *
 * ARGUMENTS:           LPSTR
            lpszQueryString                         client browser
http command string
 *
            NEW_ORDER_DATA      *pNewOrderData
 *          pointer to new order data structure
 */

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData)
{
            char        szTmp[26];
            int                     i;
            short       items;
            int                     ol_i_id, ol_quantity;
            char        *ptr = lpszQueryString;

            static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
                        { "SP00*", "SP01*", "SP02*", "SP03*",
"SP04*",
                          "SP05*", "SP06*", "SP07*", "SP08*",
"SP09*",
                          "SP10*", "SP11*", "SP12*", "SP13*",
"SP14*" };
            static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
                        { "IID00*", "IID01*", "IID02*", "IID03*",
"IID04*",
                          "IID05*", "IID06*", "IID07*", "IID08*",
"IID09*",
                          "IID10*", "IID11*", "IID12*", "IID13*",
"IID14*" };
            static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
                        { "Qty00*", "Qty01*", "Qty02*", "Qty03*",
"Qty04*",
                          "Qty05*", "Qty06*", "Qty07*", "Qty08*",
"Qty09*",
                          "Qty10*", "Qty11*", "Qty12*", "Qty13*",
"Qty14*" };

            pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
            pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

            for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)
            {
                        GetKeyValue(&ptr, szSP[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_SUPPW_KEY);
                        if ( szTmp[0] )
                        {
                                    if ( !IsNumeric(szTmp) )
                                                throw new
CWEBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
                                    pNewOrderData-
>OL[items].ol_supply_w_id = atoi(szTmp);
                                    ol_i_id = pNewOrderData-
>OL[items].ol_i_id =

                        GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_ITEMID_INVALID);
                                    if ( ol_i_id > 999999 || ol_i_id
< 1 )
                                                throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE );

                                    ol_quantity = pNewOrderData-
>OL[items].ol_quantity =

                        GetIntKeyValue(&ptr, szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_QTY_INVALID);
                                    if ( ol_quantity > 99 ||
ol_quantity < 1 )
                                                throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

                                    items++;
                        }
                        else
                        {           // nothing entered for supply
warehouse, so item id and qty must also be blank
                                    GetKeyValue(&ptr, szIID[i],
szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_IID_KEY);
                                    if ( szTmp[0] )
                                                throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

                                    GetKeyValue(&ptr, szQty[i],
szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_QTY_KEY);
                                    if ( szTmp[0] )
                                                throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
                        }
            }
            if ( items == 0 )
                        throw new CWEBCLNT_ERR(
ERR_NEWORDER_NOITEMS_ENTERED );

            pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *
 * PURPOSE: This function extracts and validates the payment form
data from an http command string.
```

```
 *
 * ARGUMENTS:		LPSTR
		lpszQueryString				client browser
http command string
 *							PAYMENT_DATA
		*pPaymentData				pointer to
payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;
		BOOL		bCustIdBlank;
		int			iLen;

		pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_DISTRICT_INVALID);

		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{
			bCustIdBlank = TRUE;
			pPaymentData->c_id = 0;
		}
		else
		{	// parse customer id and verify that last
name was NOT entered
			bCustIdBlank = FALSE;
			if ( !IsNumeric(szTmp) )
				throw new CWEBCLNT_ERR(
ERR_PAYMENT_CUSTOMER_INVALID );
			pPaymentData->c_id = atoi(szTmp);
		}

		pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
		pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*",
ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

		if ( bCustIdBlank )
		{	// customer id is blank, so last name must be
entered
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new CWEBCLNT_ERR(
ERR_PAYMENT_MISSING_CID_CLT );
			if ( strlen(szTmp) > LAST_NAME_LEN )
				throw new CWEBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );

			strcpy(pPaymentData->c_last, szTmp);
			// pad with spaces so that the client layer
doesn't have to do it
			// before passing parameters to stored
procedure
			iLen = strlen(pPaymentData->c_last);
			memset(pPaymentData->c_last + iLen, ' ',
LAST_NAME_LEN - iLen);
			pPaymentData->c_last[LAST_NAME_LEN] =
0;
		}
		else
		{	// parse customer id and verify that last
name was NOT entered
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new CWEBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
		}

		GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_HAM_KEY);
		if (!IsDecimal(szTmp))
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_HAM_INVALID );
		pPaymentData->h_amount = atof(szTmp);
		if ( pPaymentData->h_amount >= 10000.00 ||
pPaymentData->h_amount < 0 )
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_HAM_RANGE );
}

/* FUNCTION: GetOrderStatusData
 *
 * PURPOSE: This function extracts and validates the payment form
data from an http command string.
 *
 */
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;
		int			iLen;

		pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);
		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{	// customer id is blank, so last name must be
entered
			pOrderStatusData->c_id = 0;
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

			_strupr( szTmp );
			if ( strlen(szTmp) > LAST_NAME_LEN )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

			strcpy(pOrderStatusData->c_last, szTmp);
			// pad with spaces so that the client layer
doesn't have to do it
			// before passing parameters to stored
procedure
			iLen = strlen(pOrderStatusData->c_last);
			memset(pOrderStatusData->c_last + iLen, '
', LAST_NAME_LEN - iLen);
			pOrderStatusData->c_last[LAST_NAME_LEN]
= 0;
		}
		else
		{	// parse customer id and verify that last
name was NOT entered
			if ( !IsNumeric(szTmp) )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
			pOrderStatusData->c_id = atoi(szTmp);
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
		}

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE: This function determines if a string is numeric. It fails if
any characters other
 *				than numeric and null
terminator are present.
 *
 * ARGUMENTS:		char				*ptr
		pointer to string to check.
```

```
 *
 * RETURNS:                BOOL            FALSE           if string is not all
numeric
 *
                TRUE            if string contains only numeric characters i.e.
'0' - '9'
 */

BOOL IsNumeric(char *ptr)
{
                if ( *ptr == 0 )
                                return FALSE;

                while( *ptr && isdigit(*ptr) )
                                ptr++;
                return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE: This function determines if a string is a non-negative
decimal value.
 *         It fails if any characters other than a series of numbers
followed by
 *                                              a decimal point, another series
of numbers, and a null terminator are present.
 *
 * ARGUMENTS:              char                                            *ptr
                pointer to string to check.
 *
 * RETURNS:                BOOL            FALSE           if string is not a
valid non-negative decimal value
 *
                TRUE            if string is OK
 */

BOOL IsDecimal(char *ptr)
{
                char *dotptr;
                BOOL  bValid;

                if ( *ptr == 0 )
                                return FALSE;

                // find decimal point
                dotptr = strchr( ptr, '.' );
                if (dotptr == NULL)
                                // no decimal point, so just check for
numeric
                                return IsNumeric(ptr);
                *dotptr = 0;  // temporarily replace decimal with a
terminator
                if ( *ptr != 0 )
                                bValid = IsNumeric(ptr);
                // string starts with decimal point
                else if (*(dotptr+1) == 0)
                                return FALSE;  // nothing but a decimal point
is bad
                else
                                bValid = TRUE;

                if (*(dotptr+1) != 0)
                                // check text after decimal point
                                bValid &= IsNumeric(dotptr+1);

                *dotptr = '.';  // replace decimal point
                return bValid;
}


/*              FILE:                           TPCC.C
 *                                                              Microsoft TPC-C
Kit Ver. 4.20.000
 *                                                              Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                                              Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:        Main module for TPCC.DLL which is an ISAPI
service dll.
 *              Contact:        Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                              4.20.000 - reworked error handling; added
options for COM and Encina txn monitors
 */

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <malloc.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys\timeb.h>
#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#endif
#include <icapexp.h>

#include "..\..\common\src\trans.h"                     //tpckit
transaction header contains defination of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\common\txnlog\include\rtetime.h"
#include "..\..\common\txnlog\include\spinlock.h"
#include "..\..\common\txnlog\include\txnlog.h"

// Database layer includes
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  //
DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    //
ODBC implementation of TPC-C txns

// Txn monitor layer includes
#include "..\..\tm_com_dll\src\tpcc_com.h"
        // COM Services implementation on TPC-C txns
#include "..\..\tm_tuxedo_dll\src\tpcc_tux.h"                   //
interface to Tuxedo libraries
#include "..\..\tm_encina_dll\src\tpcc_enc.h"                   //
interface to Encina libraries

#include "httpext.h"
        //ISAPI DLL information header
#include "tpcc.h"
        //this dlls specific structure, value e.t. header.


#define     LEN_ERR_STRING          256

// defines for Make<Txn>Form calls to distinguish input and output
flavors
#define     OUTPUT_FORM             0
#define INPUT_FORM      1

char
        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

//Terminal client id structure
TERM        Term = { 0, 0, 0, NULL };

// The WEBCLIENT_VERSION string specifies the version level of this
web client interface.
// The RTE must be synchronized with the interface level on login,
otherwise the login
// will fail.  This is a sanity check to catch problems resulting from
mismatched versions
// of the RTE and web client.
#define WEBCLIENT_VERSION "410"
```

```
static          CRITICAL_SECTION
                TermCriticalSection;

static  HINSTANCE hLibInstanceTm = NULL;
static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB        *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;
TYPE_CTPCC_TUXEDO       *pCTPCC_TUXEDO_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_new;
TYPE_CTPCC_ENCINA   *pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM                  *pCTPCC_COM_new;

// For deferred Delivery txns:

CTxnLog
                *txnDelilog = NULL;
                //used to log delivery transaction information

HANDLE
                hWorkerSemaphore        = INVALID_HANDLE_VALUE;
HANDLE
                hDoneEvent                              =
INVALID_HANDLE_VALUE;
HANDLE
                *pDeliHandles                           = NULL;

// configuration settings from registry
TPCCREGISTRYDATA                Reg;


DWORD
                dwNumDeliveryThreads = 4;
CRITICAL_SECTION                DelBuffCriticalSection;
                //critical section for delivery transactions cache
DELIVERY_TRANSACTION *pDelBuff                                  =
NULL;
DWORD
                dwDelBuffSize                           = 100;
                // size of circular buffer for delivery txns
DWORD
                dwDelBuffFreeCount;
                // number of buffers free
DWORD
                dwDelBuffBusyIndex      = 0;                    //
index position of entry waiting to be delivered
DWORD
                dwDelBuffFreeIndex      = 0;                    //
index position of unused entry

// Critical section to synchronize connection open and close.
CRITICAL_SECTION hConnectCriticalSection;

#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: DllMain
 *
 * PURPOSE: This function is the entry point for the DLL.  This
implementation is based on the
 *                                      fact that
DLL_PROCESS_ATTACH is only called from the inet service once.
 *
 * ARGUMENTS:           HANDLE      hModule
                        module handle
 *                                              DWORD
                ul_reason_for_call      reason for call
 *                                              LPVOID
                lpReserved                      reserved for
future use
 *
 * RETURNS:             BOOL        FALSE
                        errors occured in initialization
 *
                TRUE                                            DLL
successfully initialized
 */

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call,
LPVOID lpReserved)
{
                DWORD  i;
                char szEvent[LEN_ERR_STRING] = "\0";
                char szLogFile[128];
                char szDllName[128];

// debugging....
// DebugBreak();

                try
                {
                        switch( ul_reason_for_call )
                        {
                                case DLL_PROCESS_ATTACH:
                                        {

                DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;

                GetComputerName(szMyComputerName, &dwSize);

                szMyComputerName[dwSize] = 0;
                                        }


                DisableThreadLibraryCalls((HMODULE)hModule);

                InitializeCriticalSection(&TermCriticalSection);
                                                        if (
ReadTPCCRegistrySettings( &Reg ) )

                throw new CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

                                                        dwDelBuffSize =
min( Reg.dwMaxPendingDeliveries, 10000 );  // min with 10000 as a
sanity constraint

                dwNumDeliveryThreads = min(
Reg.dwNumberOfDeliveryThreads, 100 );  // min with 100 as a sanity
constraint

                                                        TermInit();

                                                        // load DLL for
txn monitor
                                                        if (Reg.eTxnMon
== TUXEDO)
                                                        {

                strcpy( szDllName, Reg.szPath );

                strcat( szDllName, "tpcc_tuxedo.dll");

                hLibInstanceTm = LoadLibrary( szDllName );
                                                                if
(hLibInstanceTm == NULL)

                throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                                                //
get function pointer to wrapper for class constructor

                pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
                                                                if
(pCTPCC_TUXEDO_new == NULL)

                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                        }
                                                        else if
(Reg.eTxnMon == ENCINA)
                                                        {

                strcpy( szDllName, Reg.szPath );

                strcat( szDllName, "tpcc_encina.dll");
```

```
        hLibInstanceTm = LoadLibrary( szDllName );
                                                        if
(hLibInstanceTm == NULL)
        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
                                                        //
get function pointer to wrapper for class constructor
        pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");
        pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
                                                        if
(pCTPCC_ENCINA_new == NULL)
        throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                }
                                                else if
(Reg.eTxnMon == COM)
                                                {
        strcpy( szDllName, Reg.szPath );
        strcat( szDllName, "tpcc_com.dll");
        hLibInstanceTm = LoadLibrary( szDllName );
                                                        if
(hLibInstanceTm == NULL)
        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
                                                        //
get function pointer to wrapper for class constructor
        pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm,"CTPCC_COM_new");
                                                        if
(pCTPCC_COM_new == NULL)
        throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                }
                                                // load DLL for
database connection
                                                if ((Reg.eTxnMon
== None) || (dwNumDeliveryThreads > 0))
                                                        {
                                                        if
(Reg.eDB_Protocol == DBLIB)
                                                        {
        strcpy( szDllName, Reg.szPath );
        strcat( szDllName, "tpcc_dblib.dll");
        hLibInstanceDb = LoadLibrary( szDllName );
        if (hLibInstanceDb == NULL)
                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );
        // get function pointer to wrapper for class constructor
        pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
        if (pCTPCC_DBLIB_new == NULL)
                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                }
                                                else
if (Reg.eDB_Protocol == ODBC)
                                                {
        strcpy( szDllName, Reg.szPath );
        strcat( szDllName, "tpcc_odbc.dll");
        hLibInstanceDb = LoadLibrary( szDllName );
        if (hLibInstanceDb == NULL)
                throw new CWEBCLNT_ERR(
ERR_LOADDLL_FAILED, szDllName, GetLastError() );
        // get function pointer to wrapper for class constructor
        pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
        if (pCTPCC_ODBC_new == NULL)
                throw new CWEBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                                }
                                        }
                                        if
(dwNumDeliveryThreads)
                                        {
                                        //
Initialize delivery delay critical section
                                        //
        InitializeCriticalSection(&hConnectCriticalSection);
                                        //
for deferred delivery txns:
        hDoneEvent = CreateEvent( NULL, TRUE /* manual reset
*/, FALSE /* initially not signalled */, NULL );
        InitializeCriticalSection(&DelBuffCriticalSection);
        hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );
        dwDelBuffFreeCount = dwDelBuffSize;
        InitJulianTime(NULL);
                                        //
create unique log file name based on delilog-yymmdd-hhmm.log
        SYSTEMTIME Time;
        GetLocalTime( &Time );
        wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-
%2.2d%2.2d-%2.2ds%2.2dms.log",
        Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute, Time.wSecond,
Time.wMilliseconds );
        txnDelilog = new CTxnLog(szLogFile, TXN_LOG_WRITE);
        //write event into txn log for START
        txnDelilog->WriteCtrlRecToLog(TXN_EVENT_START,
szMyComputerName, sizeof(szMyComputerName));
                                        //
allocate structures for delivery buffers and thread mgmt
```

```
		pDeliHandles = new HANDLE[dwNumDeliveryThreads];

		pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
                                                                //
launch DeliveryWorkerThread to perform actual delivery txns
		for(i=0; i<dwNumDeliveryThreads; i++)
                                                                {

		pDeliHandles[i] = (HANDLE) _beginthread(
DeliveryWorkerThread, 0, NULL );

		if (pDeliHandles[i] == INVALID_HANDLE_VALUE)

                        throw new CWEBCLNT_ERR(
ERR_DELIVERY_THREAD_FAILED );
                                                                }
                                                        }

                                                        break;

                                        case DLL_PROCESS_DETACH:
                                                        if
(dwNumDeliveryThreads)
                                                        {
                                                                if
(txnDelilog != NULL)
                                                                {

		//write event into txn log for STOP

		txnDelilog->WriteCtrlRecToLog(TXN_EVENT_STOP,
szMyComputerName, sizeof(szMyComputerName));


		// This will do a clean shutdown of the delivery log file

		CTxnLog	*txnDelilogLocal = txnDelilog;

		txnDelilog= NULL;

		delete txnDelilogLocal;
                                                                }


		delete [] pDeliHandles;

		delete [] pDelBuff;

		CloseHandle( hWorkerSemaphore );
		CloseHandle( hDoneEvent );

		DeleteCriticalSection(&DelBuffCriticalSection);

                                                                //
Delete delivery delay critical section
                                                                //

		DeleteCriticalSection(&hConnectCriticalSection);
                                                        }


		DeleteCriticalSection(&TermCriticalSection);

                                                        if
(hLibInstanceTm != NULL)

		FreeLibrary( hLibInstanceTm );
                                                        hLibInstanceTm
= NULL;

                                                        if
(hLibInstanceDb != NULL)

		FreeLibrary( hLibInstanceDb );
                                                        hLibInstanceDb =
NULL;

                                                        Sleep(500);
                                                        break;

                                        default:
                                                        /* nothing */;
                        }
                }
                catch (CBaseErr *e)
                {
                        TCHAR szMsg[256];

                        _sntprintf(szMsg, sizeof(szMsg), "%s error,
code %d: %s",
                                                        e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                        WriteMessageToEventLog( szMsg );
                        delete e;
                        TerminateExtension(0);
                        return FALSE;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception.  DLL could not load.."));
                        TerminateExtension(0);
                        return FALSE;
                }

                return TRUE;
}


/* FUNCTION: GetExtensionVersion
 *
 * PURPOSE: This function is called by the inet service when the DLL is
first loaded.
 *
 * ARGUMENTS:		HSE_VERSION_INFO	*pVer
		passed in structure in which to place expected version
number.
 *
 * RETURNS:		TRUE		inet service expected return
value.
 */

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
		pVer->dwExtensionVersion =
MAKELONG(HSE_VERSION_MINOR, HSE_VERSION_MAJOR);
		lstrcpyn(pVer->lpszExtensionDesc, "TPC-C Server.",
HSE_MAX_EXT_DLL_NAME_LEN);

		// TODO: why do we need this here instead of in the DLL
attach?
		if (Reg.eTxnMon == ENCINA)
			pCTPCC_ENCINA_post_init();

		return TRUE;
}

/* FUNCTION: TerminateExtension
 *
 * PURPOSE: This function is called by the inet service when the DLL is
about to be unloaded.
 *			Release all resources in
anticipation of being unloaded.
 *
 * RETURNS:		TRUE		inet service expected return
value.
 */

BOOL WINAPI TerminateExtension( DWORD dwFlags )
{
		if (pDeliHandles)
		{
			SetEvent( hDoneEvent );
```

```
                        for(DWORD i=0; i<dwNumDeliveryThreads;
i++)
                                WaitForSingleObject(
pDeliHandles[i], INFINITE );
        }

        TermDeleteAll();
        return TRUE;
}


/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE: This function is the main entry point for the TPCC DLL.
The internet service
 *                      calls this function passing in
the http string.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
        *pECB           structure pointer to passed in internet
 *
                        service information.
 *
 * RETURNS:             DWORD   HSE_STATUS_SUCCESS
                                        connection can
be dropped if error
 *
        HSE_STATUS_SUCCESS_AND_KEEP_CONN
        keep connect valid comment sent
 *
 * COMMENTS:            None
 *
 */

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK
*pECB)
{
        int                             iCmd, FormId,
TermId, iSyncId;
        char                    szBuffer[4096];

        int                             lpbSize;
        static char     szHeader[] = "200 Ok";
        DWORD                   dwSize = 6;             //
initial value is strlen(szHeader)
        char                    szHeader1[4096];


#ifdef ICECAP
        StartCAP();
#endif

        try
        {
                //process http query
                ProcessQueryString(pECB, &iCmd, &FormId,
&TermId, &iSyncId);

                if (TermId != 0)
                {
                        if ( TermId < 0 || TermId >=
Term.iNumEntries || Term.pClientData[TermId].iNextFree != -1 )
                        {
                                // debugging...
                                char szTmp[128];
                                wsprintf( szTmp,
"Invalid term ID; TermId = %d", TermId );

        WriteMessageToEventLog( szTmp );

                                throw new
CWEBCLNT_ERR( ERR_INVALID_TERMID );
                        }

                        //must have a valid syncid here
since termid is valid
                        if (iSyncId !=
Term.pClientData[TermId].iSyncId)
                                throw new
CWEBCLNT_ERR( ERR_INVALID_SYNC_CONNECTION );

                        //set use time

        Term.pClientData[TermId].iTickCount = GetTickCount();
                }


                switch(iCmd)
                {
                case 0:
                        WelcomeForm(pECB, szBuffer);
                        break;
                case 1:
                        switch( FormId )
                        {
                                case
WELCOME_FORM:
                                case
MAIN_MENU_FORM:

        break;
                                case
NEW_ORDER_FORM:

        ProcessNewOrderForm(pECB, TermId, szBuffer);

        break;                          case
PAYMENT_FORM:

        ProcessPaymentForm(pECB, TermId, szBuffer);

        break;
                                case
DELIVERY_FORM:

        ProcessDeliveryForm(pECB, TermId, szBuffer);

        break;
                                case
ORDER_STATUS_FORM:

        ProcessOrderStatusForm(pECB, TermId, szBuffer);

        break;
                                case
STOCK_LEVEL_FORM:

        ProcessStockLevelForm(pECB, TermId, szBuffer);

        break;
                        }
                        break;
                case 2:
                        // new-order selected from
menu; display new-order input form
                        MakeNewOrderForm(TermId,
NULL, INPUT_FORM, szBuffer);
                        break;
                case 3:
                        // payment selected from
menu; display payment input form
                        MakePaymentForm(TermId,
NULL, INPUT_FORM, szBuffer);
                        break;
                case 4:
                        // delivery selected from menu;
display delivery input form
                        MakeDeliveryForm(TermId,
NULL, INPUT_FORM, szBuffer);
                        break;
                case 5:
                        // order-status selected from
menu; display order-status input form
                        MakeOrderStatusForm(TermId,
NULL, INPUT_FORM, szBuffer);
                        break;
                case 6:
```

```
                                        // stock-level selected from
menu; display stock-level input form
                                        MakeStockLevelForm(TermId,
NULL, INPUT_FORM, szBuffer);
                                        break;
                        case 7:
                                        // ExitCmd
                                        TermDelete(TermId);
                                        WelcomeForm(pECB, szBuffer);
                                        break;
                        case 8:
                                        SubmitCmd(pECB, szBuffer);
                                        break;
                        case 9:
                                        // menu
                                        MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
                                        break;
                        case 10:
                                        // CMD=Clear
                                        // resets all connections;
should only be used when no other connections are active
                                        TermDeleteAll();
                                        TermInit();
                                        WelcomeForm(pECB, szBuffer);
                                        break;
                        case 11:        // CMD=Stats
                                        StatsCmd(pECB, szBuffer);
                                        break;
                        }
                }
                catch (CBaseErr *e)
                {
                        ErrorForm( pECB, e->ErrorType(), e-
>ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
                        delete e;
                }
                catch (...)
                {
                        ErrorForm( pECB, ERR_TYPE_WEBDLL, 0,
TermId, iSyncId, "Error: Unhandled exception in Web Client.", szBuffer
);
                }

#ifdef ICECAP
                StopCAP();
#endif


                lpbSize = strlen(szBuffer);
                wsprintf(szHeader1,
                                        "Content-Type: text/html\r\n"
                                        "Content-Length: %d\r\n"
                                        "Connection: Keep-
Alive\r\n\r\n" , lpbSize);
                strcat( szHeader1, szBuffer );

                (*pECB->ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER, szHeader, (LPDWORD) &dwSize,
(LPDWORD)szHeader1);

                //finish up and keep connection
                pECB->dwHttpStatusCode = 200;
                return HSE_STATUS_SUCCESS_AND_KEEP_CONN;
}


void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL, TEXT("TPCC.DLL"));

   _stprintf(szMsg, TEXT("Error in TPCC.DLL: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
                {
      ReportEvent(hEventSource, // handle of event source
         EVENTLOG_ERROR_TYPE,  // event type
         0,                    // event category
         0,                    // event ID
         NULL,                 // current user's SID
         2,                    // strings in lpszStrings
         0,                    // no bytes of raw data
         (LPCTSTR *)lpszStrings,    // array of error strings
         NULL);                // no raw data

      (VOID) DeregisterEventSource(hEventSource);
   }
}


/* FUNCTION: DeliveryWorkerThread
 *
 * PURPOSE: This function processes deferred delivery txns.  There are
typically several
 *                                      threads running this routine.
The number of threads is determined by an entry
 *                                      read from the registry.  The
thread waits for work by waiting on semaphore.  When a delivery txn is posted,
the semaphore is released.  After processing
 *                                      the delivery txn, information is
logged to record the txn status and execution
 *                                      time.
 */

/*static*/ void DeliveryWorkerThread(void *ptr)
{
                CTPCC_BASE              *pTxn = NULL;

                DELIVERY_TRANSACTION                    delivery;
                PDELIVERY_DATA
                pDeliveryData;
                TXN_RECORD_TPCC_DELIV_DEF       txnDeliRec;

                DWORD
                index;
                HANDLE
                handles[2];

                SYSTEMTIME              trans_end;
                //delivery transaction finished time
                SYSTEMTIME              trans_start;  //delivery
transaction start time

                assert(txnDelilog != NULL);

                try
                {
                        if (Reg.eDB_Protocol == ODBC)
                        {
                                if (Reg.dwConnectDelay > 0)
                                {
                                        // Synchronize
connect (for VIA)
                                        //

                EnterCriticalSection(&hConnectCriticalSection);


                Sleep(Reg.dwConnectDelay);

                                                pTxn =
pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,


                szMyComputerName, Reg.szDbName,
```

```
                Reg.szSPPrefix, Reg.bCallNoDuplicatesNewOrder );


                LeaveCriticalSection(&hConnectCriticalSection);
                                }
                        }
                        else
                        {
                                if (Reg.eDB_Protocol ==
DBLIB)
                                        pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                        }
                        pDeliveryData = pTxn->BuffAddr_Delivery();
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "Error in Delivery Txn
thread.  Could not connect to database.  "
                                        "%s.
Server=%s, User=%s, Password=%s, Database=%s",
                                        e->ErrorText(),
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                        WriteMessageToEventLog( szTmp );
                        delete e;
                        goto ErrorExit;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread."));
                        goto ErrorExit;
                }

                while (TRUE)
                {
                        try
                        {
                                //while delivery thread running,
i.e. user has not requested termination
                                while (TRUE)
                                {
                                        // need to wait
for multiple objects: program exit or worker semaphore;
                                        handles[0] =
hDoneEvent;
                                        handles[1] =
hWorkerSemaphore;
                                        index =
WaitForMultipleObjects( 2, &handles[0], FALSE, INFINITE);
                                        if (index ==
WAIT_OBJECT_0)
                                                goto
ErrorExit;


                                ZeroMemory(&txnDeliRec, sizeof(txnDeliRec));

                                txnDeliRec.TxnType = TXN_REC_TYPE_TPCC_DELIV_DEF;

                                        // make a local
copy of current entry from delivery buffer and increment buffer index

                                EnterCriticalSection(&DelBuffCriticalSection);
                                        delivery =
*(pDelBuff+dwDelBuffBusyIndex);

                                dwDelBuffFreeCount++;

                                dwDelBuffBusyIndex++;
                                        if
(dwDelBuffBusyIndex == dwDelBuffSize)           // wrap-around if
at end of buffer

                                dwDelBuffBusyIndex = 0;


                                LeaveCriticalSection(&DelBuffCriticalSection);

                                        pDeliveryData-
>w_id = delivery.w_id;
                                        pDeliveryData-
>o_carrier_id = delivery.o_carrier_id;

                                        txnDeliRec.w_id
= pDeliveryData->w_id;

                                txnDeliRec.o_carrier_id = pDeliveryData->o_carrier_id;

                                txnDeliRec.TxnStartT0 = Get64BitTime(&delivery.queue);

                                        GetLocalTime(
&trans_start );
                                        pTxn->Delivery();
                                        GetLocalTime(
&trans_end );

                                        //log txn

                                txnDeliRec.TxnStatus = ERR_SUCCESS;
                                        for (int i=0; i<10;
i++)
                                txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
                                txnDeliRec.DeltaT4 = (int)(Get64BitTime(&trans_end) -
txnDeliRec.TxnStartT0);

                                txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                        if (txnDelilog !=
NULL)

                                txnDelilog->WriteToLog(&txnDeliRec);
                                        }
                                }
                                catch (CBaseErr *e)
                                {
                                        char szTmp[1024];
                                        wsprintf( szTmp, "%s Error
(code %d) in Delivery Txn thread. %s",
                                                e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText() );
                                        WriteMessageToEventLog(
szTmp );

                                        // log the error txn
                                        txnDeliRec.TxnStatus = e-
>ErrorType();
                                        if (txnDelilog != NULL)
                                                txnDelilog-
>WriteToLog(&txnDeliRec);

                                        delete e;
                                }
                                catch (...)
                                {
                                        // unhandled exception;
shouldn't happen; not much we can do...

                                WriteMessageToEventLog(TEXT("Unhandled exception
caught in DeliveryWorkerThread."));
                                }
                        }

ErrorExit:
                if (Reg.dwConnectDelay > 0)
                {
                        // Synchronize disconnect (for VIA)
                        //

                EnterCriticalSection(&hConnectCriticalSection);

                        Sleep(Reg.dwConnectDelay);
```

```
                delete pTxn;


        LeaveCriticalSection(&hConnectCriticalSection);
        }

        _endthread();
}

/* FUNCTION: PostDeliveryInfo
 *
 * PURPOSE: This function enters the delivery txn into the deferred
delivery buffer.
 *
 * RETURNS:         BOOL    FALSE   delivery
information posted successfully
 *
        TRUE    error cannot post delivery info
 */

BOOL PostDeliveryInfo(long w_id, short o_carrier_id)
{
        BOOL bError;

        EnterCriticalSection(&DelBuffCriticalSection);
        if (dwDelBuffFreeCount > 0)
        {
                bError = FALSE;
                (pDelBuff+dwDelBuffFreeIndex)->w_id
                = w_id;
                (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id= o_carrier_id;

        GetLocalTime(&(pDelBuff+dwDelBuffFreeIndex)->queue);

                dwDelBuffFreeCount--;
                dwDelBuffFreeIndex++;
                if (dwDelBuffFreeIndex == dwDelBuffSize)
                        dwDelBuffFreeIndex = 0;
        // wrap-around if at end of buffer
        }
        else
                // No free buffers.  Return an error, which
indicates that the delivery buffer is full.
                // Most likely, the number of delivery worker
threads needs to be increased to keep up
                // with the txn rate.
                bError = TRUE;
        LeaveCriticalSection(&DelBuffCriticalSection);

        if (!bError)
                // increment worker semaphore to wake up
a worker thread
                ReleaseSemaphore( hWorkerSemaphore, 1,
NULL );

        return bError;
}

/* FUNCTION: ProcessQueryString
 *
 * PURPOSE: This function extracts the relevent information out of the
http command passed in from
 *                                      the browser.
 *
 * COMMENTS:        If this is the initial connection i.e. client is at
welcome screen then
 *                                      there will not be
a terminal id or current form id.  If this is the case
 *                                      then the pTermid
and pFormid return values are undefined.
 */

void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int
*pCmd, int *pFormId, int *pTermId, int *pSyncId)
{
        char *ptr = pECB->lpszQueryString;
        char szBuffer[25];
        int i;

        //allowable client command strings i.e. CMD=command
        static char *szCmds[] =
        {
                "Process", "..NewOrder..", "..Payment..",
"..Delivery..", "..Order-Status..", "..Stock-Level..",
                "..Exit..", "Submit", "Menu", "Clear", "Stats",
""
        };

        *pCmd    = 0;                   // default is the
login screen
        *pTermId = 0;

        // if no params (i.e., empty query string), then return
login screen
        if (strlen(pECB->lpszQueryString) == 0)
                return;

        // parse FORMID, TERMID, and SYNCID
        *pFormId = GetIntKeyValue(&ptr, "FORMID", NO_ERR,
NO_ERR);
        *pTermId = GetIntKeyValue(&ptr, "TERMID", NO_ERR,
NO_ERR);
        *pSyncId = GetIntKeyValue(&ptr, "SYNCID", NO_ERR,
NO_ERR);
        // parse CMD
        GetKeyValue(&ptr, "CMD", szBuffer, sizeof(szBuffer),
ERR_COMMAND_UNDEFINED);

        // see which command it matches
        for(i=0; ; i++)
        {
                if (szCmds[i][0] == 0)
                        // no more; no match; return
error
                        throw new CWEBCLNT_ERR(
ERR_COMMAND_UNDEFINED );
                if ( !strcmp(szCmds[i], szBuffer) )
                {
                        *pCmd = i+1;
                        break;
                }
        }
}

/* FUNCTION: void WelcomeForm
 *
 */

void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer)
{
        char szTmp[1024];

        //welcome to tpc-c html form buffer, this is first form
client sees.
        strcpy( szBuffer,
        "<HTML><HEAD><TITLE>TPC-C Web
Client</TITLE></HEAD><BODY>"

        "<B><BIG>Microsoft TPC-C Web Client (ver
4.20)</BIG></B> <BR> <BR>"

        "<font face=\"Courier New\"><PRE>"

        "Compiled: "__DATE__", "__TIME__" <BR>"

        "Source:   "__FILE__" ("__TIMESTAMP__") <BR>"

        "</PRE></font>"

        "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"

        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
```

```
                "<INPUT TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"

                "<INPUT TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"1\">"

                "<INPUT TYPE=\"hidden\" NAME=\"TERMID\"
VALUE=\"0\">"

                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"0\">"

                "<INPUT TYPE=\"hidden\" NAME=\"VERSION\"
VALUE=\"" WEBCLIENT_VERSION "\">"
                                        );

                sprintf( szTmp,                 "Configuration Settings:
<BR><font face=\"Courier New\" color=\"blue\"><PRE>"

                "Txn Monitor            = <B>%s</B><BR>"

                "Database protocol      = <B>%s</B><BR>"

                "Max Connections        = <B>%d</B><BR>"
                                                                        "#
of Delivery Threads  = <B>%d</B><BR>"

                "Max Pending Deliveries = <B>%d</B><BR>"
                                        ,
szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
                                        Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
                strcat( szBuffer, szTmp);

                if (Reg.eTxnMon == COM)
                {
                        sprintf( szTmp,                 "COM Single Pool
= <B>%s</B><BR>",
                                        Reg.bCOM_SinglePool ? "YES" :
"NO" );
                        strcat( szBuffer, szTmp);
                }
                strcat( szBuffer, "</PRE></font>");

                if (Reg.eTxnMon == None)
                        // connection options may be specified when
not using a txn monitor
                        sprintf( szTmp,                 "Please enter
your database options for this connection:<BR>"

                "<font face=\"Courier New\" color=\"blue\"><PRE>"

                "DB Server    = <INPUT NAME=\"db_server\" SIZE=20
VALUE=\"%s\"><BR>"
                "DB User ID   = <INPUT NAME=\"db_user\" SIZE=20
VALUE=\"%s\"><BR>"

                "DB Password  = <INPUT NAME=\"db_passwd\" SIZE=20
VALUE=\"%s\"><BR>"

                "DB Name      = <INPUT NAME=\"db_name\" SIZE=20
VALUE=\"%s\"><BR>"

                "</PRE></font>"
                                                ,
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                else
                        // if using a txn monitor, connection options
are determined from registry; can't
                        // set per user.  show options fyi
                        sprintf( szTmp,                 "Database
options which will be used by the transaction monitor:<BR>"

                "<font face=\"Courier New\" color=\"blue\"><PRE>"

                "DB Server              = <B>%s</B><BR>"

                "DB User ID             = <B>%s</B><BR>"

                "DB Password            = <B>%s</B><BR>"

                "DB Name                = <B>%s</B><BR>"

                "</PRE></font>"
                                                ,
Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
                strcat( szBuffer, szTmp);

                sprintf( szTmp,                 "Please enter your Warehouse
and District for this session:<BR>"

                "<font face=\"Courier New\" color=\"blue\"><PRE>" );
                strcat( szBuffer, szTmp);
                strcat( szBuffer,               "Warehouse ID = <INPUT
NAME=\"w_id\" SIZE=6><BR>"

                "District ID  = <INPUT NAME=\"d_id\" SIZE=2><BR>"

                "</PRE></font><HR>"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Submit\">"
}               "</FORM></BODY></HTML>");

/* FUNCTION: SubmitCmd
 *
 * PURPOSE: This function allocated a new terminal id in the Term
structure array.
 *
 */

void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
                int                             iNewTerm;
                char            *ptr = pECB->lpszQueryString;

                char            szVersion[32]           = { 0 };
                char            szServer[32] = { 0 };
                char            szUser[32]              = "sa";
                char            szPassword[32]          = { 0 };
                char            szDatabase[32]          = "tpcc";

                // validate version field; the version field ensures that the
RTE is synchronized with the web client
                GetKeyValue(&ptr, "VERSION", szVersion,
sizeof(szVersion), ERR_VERSION_MISMATCH);
                if ( strcmp( szVersion, WEBCLIENT_VERSION ) )
                        throw new CWEBCLNT_ERR(
ERR_VERSION_MISMATCH );

                if (Reg.eTxnMon == None)
                {
                        // parse Server name
                        GetKeyValue(&ptr, "db_server", szServer,
sizeof(szServer), ERR_NO_SERVER_SPECIFIED);
                        // parse User name
                        GetKeyValue(&ptr, "db_user", szUser,
sizeof(szUser), NO_ERR);
                        // parse Password
                        GetKeyValue(&ptr, "db_passwd",
szPassword, sizeof(szPassword), NO_ERR);
                        // parse Database name
                        GetKeyValue(&ptr, "db_name", szDatabase,
sizeof(szDatabase), NO_ERR);
                }

                // parse warehouse ID
                int w_id = GetIntKeyValue(&ptr, "w_id",
ERR_HTML_ILL_FORMED, ERR_W_ID_INVALID);
                if ( w_id < 1 )
                        throw new CWEBCLNT_ERR(
ERR_W_ID_INVALID );

                // parse district ID
```

```
		int d_id = GetIntKeyValue(&ptr, "d_id",
ERR_HTML_ILL_FORMED, ERR_D_ID_INVALID);
		if ( d_id < 1 || d_id > 10 )
			throw new CWEBCLNT_ERR(
ERR_D_ID_INVALID );


		iNewTerm = TermAdd();

		Term.pClientData[iNewTerm].w_id = w_id;
		Term.pClientData[iNewTerm].d_id = d_id;

		try
		{
			if (Reg.eTxnMon == TUXEDO)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_TUXEDO_new();
			else if (Reg.eTxnMon == ENCINA)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_ENCINA_new();
			else if (Reg.eTxnMon == COM)

		Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new(
Reg.bCOM_SinglePool );
			else if (Reg.eDB_Protocol == ODBC)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_ODBC_new(	szServer, szUser, szPassword,
szMyComputerName,



		szDatabase, Reg.szSPPrefix,



		Reg.bCallNoDuplicatesNewOrder );
			else if (Reg.eDB_Protocol == DBLIB)

		Term.pClientData[iNewTerm].pTxn =
pCTPCC_DBLIB_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase );
		}
		catch (...)
		{
			TermDelete(iNewTerm);
			throw;			// pass exception
upward
		}
		MakeMainMenuForm(iNewTerm,
Term.pClientData[iNewTerm].iSyncId, szBuffer);
}

/* FUNCTION: StatsCmd
 *
 * PURPOSE: This function returns to the browser the total number of
active terminal ids.
 *					This routine is for
development/debugging purposes.
 *
 */

void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer)
{
		int i;
		int		iTotal;

		EnterCriticalSection(&TermCriticalSection);

		iTotal = 0;
		for(i=0; i<Term.iNumEntries; i++)
		{
			if (Term.pClientData[i].iNextFree == -1)
				iTotal++;
		}

		LeaveCriticalSection(&TermCriticalSection);

		wsprintf( szBuffer,

		"<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"
				"<BODY><B><BIG> Total
Active Connections: %d </BIG></B><BR></BODY></HTML>"
				, iTotal );
}

char *CWEBCLNT_ERR::ErrorText()
{
		static SERRORMSG errorMsgs[] =
		{
			{		ERR_COMMAND_UNDEFINED,

		"Command undefined."

						},
			{		ERR_D_ID_INVALID,

		"Invalid District ID Must be 1 to 10."

		},
			{
		ERR_DELIVERY_CARRIER_ID_RANGE,
		"Delivery Carrier ID out of range must be 1 - 10."
			{			},
		ERR_DELIVERY_CARRIER_INVALID,
		"Delivery Carrier ID invalid must be numeric 1 - 10."
				},
			{
		ERR_DELIVERY_MISSING_OCD_KEY,
		"Delivery missing Carrier ID key \"OCD*\"."

		},
			{
		ERR_DELIVERY_THREAD_FAILED,
			"Could not start delivery worker thread."

		},
			{		ERR_GETPROCADDR_FAILED,

		"Could not map proc in DLL.  GetProcAddr error.  DLL="
						},
			{		ERR_HTML_ILL_FORMED,
					"Required key
field is missing from HTML string."
			},
			{
		ERR_INVALID_SYNC_CONNECTION,
		"Invalid Terminal Sync ID."

			},
			{		ERR_INVALID_TERMID,

		"Invalid Terminal ID."

					},
			{		ERR_LOADDLL_FAILED,

		"Load of DLL failed.  DLL="

					},
			{
		ERR_MAX_CONNECTIONS_EXCEEDED,
		"No connections available.  Max Connections is probably
too low."	},
			{
		ERR_MISSING_REGISTRY_ENTRIES,
		"Required registry entries are missing.  Rerun INSTALL to
correct."	},
			{
		ERR_NEWORDER_CUSTOMER_INVALID,
		"New Order customer id invalid data type, range = 1 to
3000."			},
```

```
                {
        ERR_NEWORDER_CUSTOMER_KEY,
                "New Order missing Customer key \"CID*\"."

                },
                {
        ERR_NEWORDER_DISTRICT_INVALID,
        "New Order District ID Invalid range 1 - 10."
                                                        },
                {
        ERR_NEWORDER_FORM_MISSING_DID,
        "New Order missing District key \"DID*\"."

        },
                {
        ERR_NEWORDER_ITEMID_INVALID,
        "New Order Item Id is wrong data type, must be
numeric."                       },
                {
        ERR_NEWORDER_ITEMID_RANGE,
                "New Order Item Id is out of range. Range =
1 to 999999."                           },
                {
        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        "New Order Item_Id field entered without a corresponding
Supp_W."    },
                {
        ERR_NEWORDER_MISSING_IID_KEY,
        "New Order missing Item Id key \"IID*\"."

        },
                {
        ERR_NEWORDER_MISSING_QTY_KEY,
        "New Order Missing Qty key \"Qty##*\"."

                },
                {
        ERR_NEWORDER_MISSING_SUPPW_KEY,
                "New Order missing Supp_W key
\"SP##*\"."
                },
                {
        ERR_NEWORDER_NOITEMS_ENTERED,
        "New Order No order lines entered."

                },
                {
        ERR_NEWORDER_QTY_INVALID,
                "New Order Qty invalid must be numeric
range 1 - 99."                                          },
                {
        ERR_NEWORDER_QTY_RANGE,
                                "New Order Qty is out of
range. Range = 1 to 99."
        },      {
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                "New Order Qty field entered without a
corresponding Supp_W."          },
                {
        ERR_NEWORDER_SUPPW_INVALID,
                "New Order Supp_W invalid data type must
be numeric."                                    },
                {       ERR_NO_SERVER_SPECIFIED,
                                        "No Server name
specified."

        },
                {
        ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name may be
entered, not both."     },
                {
        ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must be numeric
1 - 3000."      },
                {
        ERR_ORDERSTATUS_CLT_RANGE,
                "Order Status Customer last name longer
than 16 characters."            },
                {
        ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be numeric 1 -
10."             },
                {
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last Name must be
entered."              },
                {
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key \"CID*\"."
                                                        },
                {
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name key \"CLT*\"."
                                                        },
                {
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key \"DID*\"."
                                                        },
                {       ERR_PAYMENT_CDI_INVALID,
                                        "Payment
Customer district invalid must be numeric."
                {       ERR_PAYMENT_CID_AND_CLT,
                                        "Payment Only
Customer ID or Last Name may be entered, not both."     },
                {
        ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."
                                                        },
                {       ERR_PAYMENT_CWI_INVALID,
                                        "Payment
Customer Warehouse invalid, must be numeric."
                },
                {
        ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1 - 10."
                                                        },
                {       ERR_PAYMENT_HAM_INVALID,
                                        "Payment Amount
invalid data type must be numeric."
                },
                {       ERR_PAYMENT_HAM_RANGE,

        "Payment Amount out of range, 0 - 9999.99."

        },
                {
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."
                                                        },
                {
        ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key \"CDI*\"."
                                                        },
                {
        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name must be
entered."                       },
                {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \"CID*\"."

        },
                {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key \"CLT*\"."
                                                                },
                {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key \"CWI*\"."
                                                        },
```

```
		{
	ERR_PAYMENT_MISSING_DID_KEY,
	"Payment missing District Key \"DID*\"."

	},
		{
	ERR_PAYMENT_MISSING_HAM_KEY,
	"Payment missing Amount key \"HAM*\"."

		},
		{
	ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
	"Stock Level; missing Threshold key \"TT*\"."
							},
		{
	ERR_STOCKLEVEL_THRESHOLD_INVALID,
	"Stock Level; Threshold value must be in the range = 1 - 99."
		},
		{
	ERR_STOCKLEVEL_THRESHOLD_RANGE,
		"Stock Level Threshold out of range, range must be 1 - 99."
							},
		{	ERR_VERSION_MISMATCH,

	"Invalid version field.  RTE and Web Client are probably out of sync." },
		{	ERR_W_ID_INVALID,

	"Invalid Warehouse ID."

							},
		{	0,

			""



	};

	char szTmp[256];
	int i = 0;
	while (TRUE)
	{
		if (errorMsgs[i].szMsg[0] == 0)
		{
			strcpy( szTmp, "Unknown error number." );
			break;
		}
		if (m_Error == errorMsgs[i].iError)
		{
			strcpy( szTmp, errorMsgs[i].szMsg );
			break;
		}
		i++;
	}

	if (m_szTextDetail)
		strcat( szTmp, m_szTextDetail );
	if (m_SystemErr)
		wsprintf( szTmp+strlen(szTmp), " Error=%d", m_SystemErr );

	m_szErrorText = new char[strlen(szTmp)+1];
	strcpy( m_szErrorText, szTmp );
	return m_szErrorText;

}

/* FUNCTION: GetKeyValue
 *
 * PURPOSE: This function parses a http formatted string for specific key values.
 *
 * ARGUMENTS:	char
	*pQueryString		http string from client browser
 *				char
			*pKey				key value to look for
 *				char
			*pValue
	character array into which to place key's value
 *					int
				iMax
	maximum length of key value array.
 *					WEBERROR
	err
	error value to throw
 *
 * RETURNS:		nothing.
 *
 * ERROR:		if (the pKey value is not found) then
 *							if (err == 0)
 *
	return (empty string)
 *							else
 *
	throw CWEBCLNT_ERR(err)
 *
 * COMMENTS:		http keys are formatted either KEY=value& or KEY=value\0. This DLL formats
 *							TPC-C input fields in such a manner that the keys can be extracted in the
 */					above manner.

void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax, WEBERROR err)
{
	char *ptr;

	if ( !(ptr=strstr(*pQueryString, pKey)) )
		goto ErrorExit;
	ptr += strlen(pKey);
	if ( *ptr != '=' )
		goto ErrorExit;
	ptr++;

	iMax--;  // one position is for terminating null
	while( *ptr && *ptr != '&' && iMax)
	{
		*pValue++ = *ptr++;
		iMax--;
	}
	*pValue = 0;  // terminating null

	*pQueryString = ptr;
	return;

ErrorExit:
	if (err != NO_ERR)
		throw new CWEBCLNT_ERR( err );
	*pValue = 0;  // return empty result string
}

/* FUNCTION: GetIntKeyValue
 *
 * PURPOSE: This function parses a http formatted string for a specific key value.
 *
 * ARGUMENTS:	char
	*pQueryString		http string from client browser
 *				char
			*pKey				key value to look for
 *					WEBERROR
	NoKeyErr		error value to throw if key not found
 *					WEBERROR
	NotIntErr		error value to throw if value not numeric
 *
 * RETURNS:		integer
 *
 * ERROR:		if (the pKey value is not found) then
 *								if (NoKeyErr != NO_ERR)
```

```
 *
            throw CWEBCLNT_ERR(err)
 *                                                          else
 *
            return 0
 *                                                  else if (non-
numeric char found) then
 *                                                          if
(NotIntErr != NO_ERR) then
 *
            throw CWEBCLNT_ERR(err)
 *                                                          else
 *
            return 0
 *
 * COMMENTS:          http keys are formatted either KEY=value&
or KEY=value\0. This DLL formats
 *                                                  TPC-C input fields
in such a manner that the keys can be extracted in the
 *                                                  above manner.
 */

int GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR
NoKeyErr, WEBERROR NotIntErr)
{
            char *ptr0;
            char *ptr;

            if ( !(ptr=strstr(*pQueryString, pKey)) )
                        goto ErrorNoKey;
            ptr += strlen(pKey);
            if ( *ptr != '=' )
                        goto ErrorNoKey;
            ptr++;

            ptr0 = ptr;                 // remember starting point
            // scan string until a terminator (null or &) or a non-digit
            while( *ptr && *ptr != '&' && isdigit(*ptr) )
                        ptr++;

            // make sure we stopped scanning for the right reason
            if ((ptr0 == ptr) || (*ptr && *ptr != '&'))
            {
                        if (NotIntErr != NO_ERR)
                                    throw new CWEBCLNT_ERR(
NoKeyErr );
                        return 0;
            }

            *pQueryString = ptr;
            return atoi(ptr0);

ErrorNoKey:  if (NoKeyErr != NO_ERR)
                        throw new CWEBCLNT_ERR( NoKeyErr );
            return 0;
}

/* FUNCTION: TermInit
 *
 * PURPOSE: This function initializes the client terminal structure; it is
called when the TPCC.DLL
 *                                          is first loaded by the inet
service.
 *
 */

void TermInit(void)
{
            EnterCriticalSection(&TermCriticalSection);

            Term.iMasterSyncId      = 1;
            Term.iNumEntries        = Reg.dwMaxConnections+1;

            Term.pClientData        = NULL;
            Term.pClientData        =
(PCLIENTDATA)malloc(Term.iNumEntries * sizeof(CLIENTDATA));
            if (Term.pClientData == NULL)
            {
                        LeaveCriticalSection(&TermCriticalSection);
                        throw new CWEBCLNT_ERR(
ERR_MEM_ALLOC_FAILED );
            }

            ZeroMemory( Term.pClientData, Term.iNumEntries *
sizeof(CLIENTDATA) );

            Term.iFreeList                      =
Term.iNumEntries-1;
            // build free list
            // note: Term.pClientData[0].iNextFree gets set to -1,
which marks it as "in use".
            //      This is intentional, as the zero entry is used as an
anchor and never
            //                          allocated as an actual
terminal.
            for(int i=0; i<Term.iNumEntries; i++)
                        Term.pClientData[i].iNextFree = i-1;

            LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermDeleteAll
 *
 * PURPOSE: This function frees allocated resources associated with the
terminal structure.
 *
 * ARGUMENTS:          none
 *
 * RETURNS:            None
 *
 * COMMENTS:           This function is called only when the inet
service unloads the TPCC.DLL
 *
 */

void TermDeleteAll(void)
{
            EnterCriticalSection(&TermCriticalSection);

            for(int i=1; i<Term.iNumEntries; i++)
            {
                        if (Term.pClientData[i].iNextFree == -1)
                                    delete
Term.pClientData[i].pTxn;
            }

            Term.iFreeList                      = 0;
            Term.iNumEntries        = 0;
            if ( Term.pClientData )
                        free(Term.pClientData);
            Term.pClientData        = NULL;

            LeaveCriticalSection(&TermCriticalSection);
}

/* FUNCTION: TermAdd
 *
 * PURPOSE: This function assigns a terminal id which is used to
identify a client browser.
 *
 * RETURNS:            int                 assigned terminal
id
 *
 */

int TermAdd(void)
{
            DWORD       i;
            int                 iNewTerm, iTickCount;

            if (Term.iNumEntries == 0)
                        return -1;

            EnterCriticalSection(&TermCriticalSection);
            if (Term.iFreeList != 0)
            {           // position is available
                        iNewTerm = Term.iFreeList;
                        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;
```

```
                        Term.pClientData[iNewTerm].iNextFree = -
1;  // indicates this position is in use
                }
                else
                {               // no open slots, so find the slot that hasn't
been used in the longest time and reuse it
                        for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
                        {
                                if (iTickCount >
Term.pClientData[i].iTickCount)
                                {
                                        iTickCount =
Term.pClientData[i].iTickCount;
                                        iNewTerm = i;
                                }
                        }
                        // if oldest term is less than one minute old,
it probably means that more connections
                        // are being attempted than were specified
as "Max Connections" at install.  In this case,
                        // do not bump existing connection; instead,
return error to requestor.
                        if ((GetTickCount() - iTickCount) < 60000)
                        {

                LeaveCriticalSection(&TermCriticalSection);
                                throw new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
                        }
                }

                Term.pClientData[iNewTerm].iTickCount =
GetTickCount();
                Term.pClientData[iNewTerm].iSyncId =
Term.iMasterSyncId++;
                Term.pClientData[iNewTerm].pTxn = NULL;

                LeaveCriticalSection(&TermCriticalSection);
                return iNewTerm;
}

/* FUNCTION: TermDelete
 *
 * PURPOSE: This function makes a terminal entry in the Term array
available for reuse.
 *
 * ARGUMENTS:             int
                                id
                Terminal id of client exiting
 *
 */

void TermDelete(int id)
{
                if ( id > 0 && id < Term.iNumEntries )
                {
                        delete Term.pClientData[id].pTxn;

                        // put onto free list
                        EnterCriticalSection(&TermCriticalSection);

                        Term.pClientData[id].iNextFree =
Term.iFreeList;
                        Term.iFreeList = id;

                        LeaveCriticalSection(&TermCriticalSection);
                }
}


/* FUNCTION: MakeErrorForm
 */

void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szBuffer )
{
                wsprintf(szBuffer,
                        "<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<BOLD>An Error
Occurred</BOLD><BR><BR>"
                        "%s"
                        "<BR><BR><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"        "</FORM></BODY></HTML>"
                        , iType, iErrorNum, MAIN_MENU_FORM,
iTermId, iSyncId, szErrorText );
}

/* FUNCTION: MakeMainMenuForm
 */

void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm)
{
                wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Main
Menu</TITLE></HEAD><BODY>"
                        "Select Desired Transaction.<BR><HR>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
                        "</FORM></BODY></HTML>"
                        , MAIN_MENU_FORM, iTermId, iSyncId);
}

/* FUNCTION: MakeStockLevelForm
 *
 * PURPOSE: This function constructs the Stock Level HTML page.
 *
 * COMMENTS:              The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */
```

```
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C Stock
Level</TITLE></HEAD><FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Stock-Level<BR>"
                        "Warehouse: %6.6d   District: %2.2d<BR>
<BR>",
                        STOCK_LEVEL_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
                        Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

        if ( bInput )
        {
                strcpy(szForm+c,
                        "Stock Level Threshold:
<INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
                        "low stock:    </font><BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR>
<BR> <BR> <BR></PRE><HR>"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                        "</FORM></HTML>" );
        }
        else
        {
                wsprintf(szForm+c,
                        "Stock Level Threshold:
%2.2d<BR> <BR>"
                        "low stock: %3.3d</font>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></PRE><HR>"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                        "</FORM></HTML>"
                        , pStockLevelData->threshold,
pStockLevelData->low_stock);
        }
}

/* FUNCTION: MakeNewOrderForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                      be freed except
when the client terminal id is no longer needed.
 */

void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm)
{
        int                     i, c;
        BOOL            bValid;
        static          char szBR[] = " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR>";

        if (!bInput)
                assert( pNewOrderData->exec_status_code
== eOK || pNewOrderData->exec_status_code == eInvalidItem );

        bValid = (bInput || (pNewOrderData->exec_status_code
== eOK));

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
New Order<BR>"
                        , bValid ? 0 : ERR_BAD_ITEM_ID,
NEW_ORDER_FORM, iTermId, Term.pClientData[iTermId].iSyncId);

        if ( bInput )
        {
                c += wsprintf(szForm+c, "Warehouse:
%6.6d   ", Term.pClientData[iTermId].w_id );

                strcpy( szForm+c,
                        "District: <INPUT
NAME=\"DID*\" SIZE=1>                   Date:<BR>"
                        "Customer:  <INPUT
NAME=\"CID*\" SIZE=4>   Name:                   Credit:
%Disc:<BR>"
                        "Order Number:
Number of Lines:        W_tax:          D_tax:<BR> <BR>"
                        " Supp_W  Item_Id  Item
Name                Qty  Stock  B/G  Price    Amount<BR>"
                        "  <INPUT NAME=\"SP00*\"
SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP01*\"
SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP02*\"
SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP03*\"
SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP04*\"
SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP05*\"
SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP06*\"
SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP07*\"
SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
                        "  <INPUT NAME=\"SP08*\"
SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
```

```
                                        "  <INPUT NAME=\"SP09*\"
SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
                                        "  <INPUT NAME=\"SP10*\"
SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
                                        "  <INPUT NAME=\"SP11*\"
SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
                                        "  <INPUT NAME=\"SP12*\"
SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
                                        "  <INPUT NAME=\"SP13*\"
SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
                                        "  <INPUT NAME=\"SP14*\"
SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
                                        "Execution Status:
Total:<BR>"
                                        "</font></PRE><HR>"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
                                        "</FORM></HTML>"
                                        );
                }
                else
                {
                        c += wsprintf(szForm+c, "Warehouse:
%6.6d   District: %2.2d                 Date: ",
                                        pNewOrderData->w_id,
                                        pNewOrderData->d_id);

                        if ( bValid )
                        {
                                c += wsprintf(szForm+c,
"%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
                                                pNewOrderData-
>o_entry_d.day,
                                                pNewOrderData-
>o_entry_d.month,
                                                pNewOrderData-
>o_entry_d.year,
                                                pNewOrderData-
>o_entry_d.hour,
                                                pNewOrderData-
>o_entry_d.minute,
                                                pNewOrderData-
>o_entry_d.second);
                        }
                        c += wsprintf(szForm+c, "<BR>Customer:
%4.4d   Name: %-16s   Credit: %-2s   ",
                                        pNewOrderData->c_id,
pNewOrderData->c_last, pNewOrderData->c_credit);

                        if ( bValid )
                        {
                                c += sprintf(szForm+c,

                "%%Disc: %5.2f          <BR>"

                "Order Number: %8.8d  Number of Lines: %2.2d
W_tax: %5.2f   D_tax: %5.2f  <BR> <BR>"

                " Supp_W  Item_Id  Item Name               Qty  Stock
B/G  Price    Amount<BR>",

                100.0*pNewOrderData->c_discount,
                                                pNewOrderData-
>o_id,
                                                pNewOrderData-
>o_ol_cnt,
                                                100.0 *
pNewOrderData->w_tax,
                                                100.0 *
pNewOrderData->d_tax);

                                        for(i=0; i<pNewOrderData-
>o_ol_cnt; i++)
                                        {
                                                c +=
sprintf(szForm+c, "%6.6d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s
$%6.2f  $%7.2f  <BR>",

                pNewOrderData->OL[i].ol_supply_w_id,

                pNewOrderData->OL[i].ol_i_id,

                pNewOrderData->OL[i].ol_i_name,

                pNewOrderData->OL[i].ol_quantity,

                pNewOrderData->OL[i].ol_stock,

                pNewOrderData->OL[i].ol_brand_generic,

                pNewOrderData->OL[i].ol_i_price,

                pNewOrderData->OL[i].ol_amount );
                                        }
                        }
                        else
                        {
                                c += wsprintf(szForm+c, "%%Disc: %5.2f
                                                "Order Number:
%8.8d  Number of Lines:         W_tax:        D_tax:<BR> <BR>"
                                                " Supp_W
Item_Id  Item Name              Qty  Stock  B/G  Price    Amount<BR>"
                                                ,
pNewOrderData->o_id);

                                i = 0;
                        }

                        strncpy( szForm+c, szBR, (15-i)*5 );
                        c += (15-i)*5;

                        if ( bValid )
                                c += sprintf(szForm+c,
"Execution Status: Transaction commited.                Total:  $%8.2f
",
                                                pNewOrderData-
>total_amount);
                        else
                                c += wsprintf(szForm+c,
"Execution Status: Item number is not valid.            Total:");

                        strcpy(szForm+c,
                                " <BR></font></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                                "</FORM></HTML>"
                                );
                }
}

/* FUNCTION: MakePaymentForm
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                      be freed except
when the client terminal id is no longer needed.
 */

void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData,
BOOL bInput, char *szForm)
```

```
{
        int c;

        c = wsprintf(szForm,
                "<HTML><HEAD><TITLE>TPC-C
Payment</TITLE></HEAD><BODY>"
                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                "<PRE><font face=\"Courier\">
Payment<BR>"
                "Date: "
                , PAYMENT_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

        if ( !bInput )
        {
                c += wsprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
                        pPaymentData->h_date.day,
                        pPaymentData-
>h_date.month,
                        pPaymentData->h_date.year,
                        pPaymentData->h_date.hour,
                        pPaymentData-
>h_date.minute,
                        pPaymentData-
>h_date.second);
        }

        if ( bInput )
        {
                c += wsprintf(szForm+c,
                        "<BR> <BR>Warehouse:
%6.6d"
                        "                District:
<INPUT NAME=\"DID*\" SIZE=1><BR> <BR> <BR> <BR> <BR>"
                        "Customer: <INPUT
NAME=\"CID*\" SIZE=4>"
                        "Cust-Warehouse: <INPUT
NAME=\"CWI*\" SIZE=4>  "
                        "Cust-District: <INPUT
NAME=\"CDI*\" SIZE=1><BR>"
                        "Name:
<INPUT NAME=\"CLT*\" SIZE=16>      "          Since:<BR>"
Credit:<BR>"
                        "
Disc:<BR>"
                        "
Phone:<BR> <BR>"
                        "Amount Paid:          $<INPUT
NAME=\"HAM*\" SIZE=7>      New Cust-Balance:<BR>"
                        "Credit Limit:<BR> <BR>Cust-
Data: <BR> <BR> <BR> <BR> <BR></font></PRE><HR>"
                        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"

                "</BODY></FORM></HTML>"
                        ,
Term.pClientData[iTermId].w_id);
        }
        else
        {
                c += wsprintf(szForm+c,
                        "<BR> <BR>Warehouse:
%6.6d                District: %2.2d<BR>"
                        "%-20s                %-
20s<BR>"
                        "%-20s                %-
20s<BR>"
                        "%-20s %-2s %5.5s-%4.4s
%-20s %-2s %5.5s-%4.4s<BR> <BR>"
                        "Customer: %4.4d  Cust-
Warehouse: %6.6d  Cust-District: %2.2d<BR>"
                        "Name:   %-16s %-2s %-16s
Since:  %2.2d-%2.2d-%4.4d<BR>"
                        "       %-20s
Credit: %-2s<BR>"

                        ,
Term.pClientData[iTermId].w_id, pPaymentData->d_id
                        , pPaymentData->w_street_1,
pPaymentData->d_street_1
                        , pPaymentData->w_street_2,
pPaymentData->d_street_2
                        , pPaymentData->w_city,
pPaymentData->w_state, pPaymentData->w_zip, pPaymentData-
>w_zip+5
                        , pPaymentData->d_city,
pPaymentData->d_state, pPaymentData->d_zip, pPaymentData-
>d_zip+5
                        , pPaymentData->c_id,
pPaymentData->c_w_id,   pPaymentData->c_d_id
                        , pPaymentData->c_first,
pPaymentData->c_middle, pPaymentData->c_last,
pPaymentData->c_since.month,    pPaymentData->c_since.day,
                        pPaymentData->c_since.year
                        , pPaymentData->c_street_1,
pPaymentData->c_credit
                        );

        c += sprintf(szForm+c,
                        "       %-20s
%%Disc:  %5.2f<BR>",
                        pPaymentData->c_street_2,
100.0*pPaymentData->c_discount);

                c += wsprintf(szForm+c,
                        "       %-20s %-2s %5.5s-
%4.4s      Phone:  %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
                        pPaymentData->c_city,
pPaymentData->c_state, pPaymentData->c_zip, pPaymentData-
>c_zip+5,
                        pPaymentData->c_phone,
pPaymentData->c_phone+6, pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );

                c += sprintf(szForm+c,
                        "Amount Paid:          $%7.2f
New Cust-Balance: $%14.2f<BR>"
                        "Credit Limit:   $%13.2f<BR>
<BR>"
                        , pPaymentData->h_amount,
pPaymentData->c_balance
                        , pPaymentData->c_credit_lim
                        );

                if ( pPaymentData->c_credit[0] == 'B' &&
pPaymentData->c_credit[1] == 'C' )
                        c += wsprintf(szForm+c,

                "Cust-Data: %-50.50s<BR>          %-50.50s<BR>
%-50.50s<BR>          %-50.50s<BR>",

                pPaymentData->c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData->c_data+150 );
                else
                        strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");

                strcat(szForm,                "
<BR></font></PRE><HR>"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
```

```
                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"

                "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>");
                }

}

/* FUNCTION: MakeOrderStatusForm
 *
 * COMMENTS:                The internal client buffer is created when the
terminal id is assigned and should not
 *                                                                  be freed except
when the client terminal id is no longer needed.
 */

void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA
*pOrderStatusData, BOOL bInput, char *szForm)
{
                int                             i, c;
                static char szBR[] = " <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>";

                c = wsprintf(szForm,
                                "<HTML><HEAD><TITLE>TPC-C Order-
Status</TITLE></HEAD><BODY>"
                                "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                                "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                                "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                                "<PRE><font face=\"Courier\">
Order-Status<BR>"               "Warehouse: %6.6d   ",
                                ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

                if ( bInput )
                {
                                strcpy(szForm+c,
                                                "District: <INPUT
NAME=\"DID*\" SIZE=1><BR>"
                                                "Customer: <INPUT
NAME=\"CID*\" SIZE=4>   Name:                   <INPUT
NAME=\"CLT*\" SIZE=23><BR>"
                                                "Cust-Balance:<BR> <BR>"
                                                "Order-Number:          Entry-
Date:                   Carrier-Number:<BR>"
                                                "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR> <BR> <BR> <BR> <BR>"
                                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
                                                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"

                "</BODY></FORM></HTML>" );
                }
                else
                {
                                c += wsprintf(szForm+c,
                                                "District: %2.2d<BR>"
                                                "Customer: %4.4d   Name: %-
16s %-2s %-16s<BR>",
                                                pOrderStatusData->d_id,
pOrderStatusData->c_id,
                                                pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData->c_last);

                                c += sprintf(szForm+c, "Cust-Balance:
$%9.2f<BR> <BR>",
                                                pOrderStatusData-
>c_balance);

                                c += wsprintf(szForm+c,
                                                "Order-Number: %8.8d   Entry-
Date: %2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d   Carrier-Number:
%2.2d<BR>"
                                                "Supply-W     Item-Id    Qty
Amount      Delivery-Date<BR>",
                                                pOrderStatusData->o_id,
                                                pOrderStatusData-
>o_entry_d.day,
                                                pOrderStatusData-
>o_entry_d.month,
                                                pOrderStatusData-
>o_entry_d.year,                pOrderStatusData-
>o_entry_d.hour,
                                                pOrderStatusData-
>o_entry_d.minute,
                                                pOrderStatusData-
>o_entry_d.second,
                                                pOrderStatusData-
>o_carrier_id);

                                for(i=0; i< pOrderStatusData->o_ol_cnt;
i++)
                                {
                                                c += sprintf(szForm+c, "
%6.6d     %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-%4.4d<BR>",

                pOrderStatusData->OL[i].ol_supply_w_id,

                pOrderStatusData->OL[i].ol_i_id,

                pOrderStatusData->OL[i].ol_quantity,

                pOrderStatusData->OL[i].ol_amount,

                pOrderStatusData->OL[i].ol_delivery_d.day,

                pOrderStatusData->OL[i].ol_delivery_d.month,

                pOrderStatusData->OL[i].ol_delivery_d.year);
                                }

                                strncpy( szForm+c, szBR, (15-i)*5 );
                                c += (15-i)*5;

                                strcpy(szForm+c,
                                                "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>" );
                }
}

/* FUNCTION: MakeDeliveryForm
```

```
 *
 * COMMENTS:            The internal client buffer is created when the
terminal id is assigned and should not
 *                                                   be freed except
when the client terminal id is no longer needed.
 */

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData,
BOOL bInput, char *szForm)
{
        int             c;

        c = wsprintf(szForm,
                        "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"
                        "<FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"0\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"TERMID\" VALUE=\"%d\">"
                        "<INPUT TYPE=\"hidden\"
NAME=\"SYNCID\" VALUE=\"%d\">"
                        "<PRE><font face=\"Courier\">
Delivery<BR>"
                        "Warehouse: %6.6d<BR> <BR>",
                        (!bInput && (pDeliveryData-
>exec_status_code != eOK)) ? ERR_TYPE_DELIVERY_POST : 0,
                        DELIVERY_FORM, iTermId,
Term.pClientData[iTermId].iSyncId, Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy( szForm+c,
                                "Carrier Number: <INPUT
NAME=\"OCD*\" SIZE=1><BR> <BR>"
                                "Execution Status: <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE><HR>"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"

        "</BODY></FORM></HTML>" );
        }
        else
        {       wsprintf( szForm+c,
                                "Carrier Number: %2.2d<BR>
<BR>"
                                "Execution Status: %s <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                                " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> </font></PRE>"
                                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"

                "</BODY></FORM></HTML>"

                                , pDeliveryData->o_carrier_id,
                                (pDeliveryData-
>exec_status_code == eOK) ? "Delivery has been queued." : "Delivery
Post Failed     "
                                );
        }
}

/* FUNCTION: ProcessNewOrderForm
 *
 * PURPOSE: This function gets and validates the input data from the
new order form
 *                                   filling in the required input
variables. it then calls the SQLNewOrder
 *                                   transaction, constructs the
output form and writes it back to client
 *                                   browser.
 */

void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
        *pECB, int iTermId, char *szBuffer)
{
        PNEW_ORDER_DATA                 pNewOrder;

        pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();

        ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
        pNewOrder->w_id = Term.pClientData[iTermId].w_id;
        GetNewOrderData(pECB->lpszQueryString, pNewOrder);
        Term.pClientData[iTermId].pTxn->NewOrder();
        pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();
        MakeNewOrderForm(iTermId, pNewOrder,
OUTPUT_FORM, szBuffer );
}

/* FUNCTION: void ProcessPaymentForm
 *
 * PURPOSE: This function gets and validates the input data from the
payment form
 *                                   filling in the required input
variables. It then calls the SQLPayment
 *                                   transaction, constructs the
output form and writes it back to client
 *                                   browser.
 *
 * ARGUMENTS:           EXTENSION_CONTROL_BLOCK
        *pECB           passed in structure pointer from inetsrv.
 *                                                   int

        iTermId         client browser terminal id
 *
 */

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
        PPAYMENT_DATA           pPayment;

        pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
        ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
        pPayment->w_id = Term.pClientData[iTermId].w_id;
        GetPaymentData(pECB->lpszQueryString, pPayment);

        Term.pClientData[iTermId].pTxn->Payment();

        pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
        MakePaymentForm(iTermId, pPayment, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
 *
 * PURPOSE: This function gets and validates the input data from the
Order Status
 *                                   form filling in the required
input variables. It then calls the
 *                                   SQLOrderStatus transaction,
constructs the output form and writes it
 *                                   back to client browser.
```

```
 *
 * ARGUMENTS:            EXTENSION_CONTROL_BLOCK
            *pECB       passed in structure pointer from inetsrv.
 *                                                  int

            iTermId     client browser terminal id
 *
 */

void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
            PORDER_STATUS_DATA    pOrderStatus;

            pOrderStatus = Term.pClientData[iTermId].pTxn-
>BuffAddr_OrderStatus();
            ZeroMemory(pOrderStatus,
sizeof(ORDER_STATUS_DATA));
            pOrderStatus->w_id = Term.pClientData[iTermId].w_id;
            GetOrderStatusData(pECB->lpszQueryString,
pOrderStatus);

            Term.pClientData[iTermId].pTxn->OrderStatus();

            pOrderStatus = Term.pClientData[iTermId].pTxn-
>BuffAddr_OrderStatus();
            MakeOrderStatusForm(iTermId, pOrderStatus,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessDeliveryForm
 *
 * PURPOSE: This function gets and validates the input data from the
delivery form
 *                                  filling in the required input
variables. It then calls the PostDeliveryInfo
 *                                  Api, The client is then informed
that the transaction has been posted.
 *
 * ARGUMENTS:            EXTENSION_CONTROL_BLOCK
            *pECB       passed in structure pointer from inetsrv.
 *                                                  int

            iTermId     client browser terminal id
 *
 */

void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
            char        *ptr = pECB->lpszQueryString;

            PDELIVERY_DATA          pDelivery;
            pDelivery = Term.pClientData[iTermId].pTxn-
>BuffAddr_Delivery();
            ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
            pDelivery->w_id = Term.pClientData[iTermId].w_id;

            pDelivery->o_carrier_id     = GetIntKeyValue(&ptr,
"OCD*", ERR_DELIVERY_MISSING_OCD_KEY,
ERR_DELIVERY_CARRIER_INVALID);
            if ( pDelivery->o_carrier_id > 10 || pDelivery-
>o_carrier_id < 1 )
                        throw new CWEBCLNT_ERR(
ERR_DELIVERY_CARRIER_ID_RANGE );

            if (dwNumDeliveryThreads)
            {
                        //post delivery info
                        if ( PostDeliveryInfo(pDelivery->w_id,
pDelivery->o_carrier_id) )
                                    pDelivery->exec_status_code
= eDeliveryFailed;
                        else
                                    pDelivery->exec_status_code
= eOK;
            }
            else // delivery is done synchronously if no delivery
threads configured
                        Term.pClientData[iTermId].pTxn-
>Delivery();

            pDelivery = Term.pClientData[iTermId].pTxn-
>BuffAddr_Delivery();
            MakeDeliveryForm(iTermId, pDelivery, OUTPUT_FORM,
szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
 *
 * PURPOSE: This function gets and validates the input data from the
Stock Level
 *                                  form filling in the required
input variables. It then calls the
 *                                  SQLStockLevel transaction,
constructs the output form and writes it
 *                                  back to client browser.
 *
 * ARGUMENTS:            EXTENSION_CONTROL_BLOCK
            *pECB       passed in structure pointer from inetsrv.
 *                                                  int

            iTermId     client browser terminal id
 *
 */

void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int
iTermId, char *szBuffer)
{
            char                    *ptr = pECB->lpszQueryString;

            PSTOCK_LEVEL_DATA       pStockLevel;

            pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
            ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

            pStockLevel->w_id = Term.pClientData[iTermId].w_id;
            pStockLevel->d_id = Term.pClientData[iTermId].d_id;

            pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);
            if ( pStockLevel->threshold >= 100 || pStockLevel-
>threshold < 0 )
                        throw new CWEBCLNT_ERR(
ERR_STOCKLEVEL_THRESHOLD_RANGE );

            Term.pClientData[iTermId].pTxn->StockLevel();

            pStockLevel = Term.pClientData[iTermId].pTxn-
>BuffAddr_StockLevel();
            MakeStockLevelForm(iTermId, pStockLevel,
OUTPUT_FORM, szBuffer);
}

/* FUNCTION: GetNewOrderData
 *
 * PURPOSE: This function extracts and validates the new order form
data from an http command string.
 *
 * ARGUMENTS:            LPSTR
            lpszQueryString                 client browser
http command string
 *
            NEW_ORDER_DATA          *pNewOrderData
            pointer to new order data structure
 *
 */

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA
*pNewOrderData)
{
            char            szTmp[26];
            int                     i;
            short           items;
            int                     ol_i_id, ol_quantity;
            char            *ptr = lpszQueryString;
```

```
	static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
			{ "SP00*", "SP01*", "SP02*", "SP03*",
"SP04*",
			  "SP05*", "SP06*", "SP07*", "SP08*",
"SP09*",
			  "SP10*", "SP11*", "SP12*", "SP13*",
"SP14*" };
	static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "IID00*", "IID01*", "IID02*", "IID03*",
"IID04*",
			  "IID05*", "IID06*", "IID07*", "IID08*",
"IID09*",
			  "IID10*", "IID11*", "IID12*", "IID13*",
"IID14*" };
	static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
			{ "Qty00*", "Qty01*", "Qty02*", "Qty03*",
"Qty04*",
			  "Qty05*", "Qty06*", "Qty07*", "Qty08*",
"Qty09*",
			  "Qty10*", "Qty11*", "Qty12*", "Qty13*",
"Qty14*" };

	pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID,
ERR_NEWORDER_DISTRICT_INVALID);
	pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY,
ERR_NEWORDER_CUSTOMER_INVALID);

	for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)
	{
		GetKeyValue(&ptr, szSP[i], szTmp,
sizeof(szTmp), ERR_NEWORDER_MISSING_SUPPW_KEY);
		if ( szTmp[0] )
		{
			if ( !IsNumeric(szTmp) )
				throw new
CWEBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
			pNewOrderData-
>OL[items].ol_supply_w_id = atoi(szTmp);

			ol_i_id = pNewOrderData-
>OL[items].ol_i_id =

	GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_ITEMID_INVALID);
			if ( ol_i_id > 999999 || ol_i_id
< 1 )
				throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE );

			ol_quantity = pNewOrderData-
>OL[items].ol_quantity =

	GetIntKeyValue(&ptr, szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY,
ERR_NEWORDER_QTY_INVALID);
			if ( ol_quantity > 99 ||
ol_quantity < 1 )
				throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

			items++;
		}
		else
		{	// nothing entered for supply
warehouse, so item id and qty must also be blank
			GetKeyValue(&ptr, szIID[i],
szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_IID_KEY);
			if ( szTmp[0] )
				throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

			GetKeyValue(&ptr, szQty[i],
szTmp, sizeof(szTmp), ERR_NEWORDER_MISSING_QTY_KEY);
			if ( szTmp[0] )
				throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
		}
	}
	if ( items == 0 )
		throw new CWEBCLNT_ERR(
ERR_NEWORDER_NOITEMS_ENTERED );

	pNewOrderData->o_ol_cnt = items;
}

/* FUNCTION: GetPaymentData
 *
 * PURPOSE: This function extracts and validates the payment form
data from an http command string.
 *
 * ARGUMENTS:	LPSTR
	lpszQueryString			client browser
http command string
 *					PAYMENT_DATA
	*pPaymentData			pointer to
payment data structure
 */

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA
*pPaymentData)
{
	char	szTmp[26];
	BOOL	bCustIdBlank;
	char	*ptr = lpszQueryString;
	int		iLen;

	pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_DISTRICT_INVALID);

	GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
	if ( szTmp[0] == 0 )
	{
		bCustIdBlank = TRUE;
		pPaymentData->c_id = 0;
	}
	else
	{	// parse customer id and verify that last
name was NOT entered
		bCustIdBlank = FALSE;
		if ( !IsNumeric(szTmp) )
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_CUSTOMER_INVALID );
		pPaymentData->c_id = atoi(szTmp);
	}

	pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
	pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*",
ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

	if ( bCustIdBlank )
	{	// customer id is blank, so last name must be
entered
		GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
		if ( szTmp[0] == 0 )
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_MISSING_CID_CLT );

		_strupr( szTmp );
		if ( strlen(szTmp) > LAST_NAME_LEN )
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_LAST_NAME_TO_LONG );

		strcpy(pPaymentData->c_last, szTmp);
		// pad with spaces so that the client layer
doesn't have to do it
		// before passing parameters to stored
procedure
		iLen = strlen(pPaymentData->c_last);
		memset(pPaymentData->c_last + iLen, ' ',
LAST_NAME_LEN - iLen);
		pPaymentData->c_last[LAST_NAME_LEN] =
0;
```

```
		}
		else
		{		// parse customer id and verify that last
name was NOT entered
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new CWEBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
		}

		GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_HAM_KEY);
		if (!IsDecimal(szTmp))
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_HAM_INVALID );
		pPaymentData->h_amount = atof(szTmp);
		if ( pPaymentData->h_amount >= 10000.00 ||
pPaymentData->h_amount < 0 )
			throw new CWEBCLNT_ERR(
ERR_PAYMENT_HAM_RANGE );
}

/* FUNCTION: GetOrderStatusData
 *
 * PURPOSE: This function extracts and validates the payment form
data from an http command string.
 *
 */
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData)
{
		char		szTmp[26];
		char		*ptr = lpszQueryString;
		int			iLen;

		pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

		GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
		if ( szTmp[0] == 0 )
		{		// customer id is blank, so last name must be
entered
			pOrderStatusData->c_id = 0;
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] == 0 )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );
			if ( strlen(szTmp) > LAST_NAME_LEN )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

			strcpy(pOrderStatusData->c_last, szTmp);
			// pad with spaces so that the client layer
doesn't have to do it
			// before passing parameters to stored
procedure
			iLen = strlen(pOrderStatusData->c_last);
			memset(pOrderStatusData->c_last + iLen, '
', LAST_NAME_LEN - iLen);
			pOrderStatusData->c_last[LAST_NAME_LEN]
= 0;
		}
		else
		{		// parse customer id and verify that last
name was NOT entered
			if ( !IsNumeric(szTmp) )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
			pOrderStatusData->c_id = atoi(szTmp);
			GetKeyValue(&ptr, "CLT*", szTmp,
sizeof(szTmp), ERR_ORDERSTATUS_MISSING_CLT_KEY);
			if ( szTmp[0] != 0 )
				throw new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
		}

}

/* FUNCTION: BOOL IsNumeric(char *ptr)
 *
 * PURPOSE: This function determines if a string is numeric. It fails if
any characters other
 *		than numeric and null
terminator are present.
 *
 * ARGUMENTS:		char		*ptr
		pointer to string to check.
 *
 * RETURNS:		BOOL		FALSE		if string is not all
numeric
 *
		TRUE		if string contains only numeric characters i.e.
'0' - '9'
 */

BOOL IsNumeric(char *ptr)
{
		if ( *ptr == 0 )
			return FALSE;
		while( *ptr && isdigit(*ptr) )
			_strdec( szTmp, ptr );
		return ( !*ptr );
}

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE: This function determines if a string is a non-negative
decimal value.
 *		It fails if any characters other than a series of numbers
followed by
 *		a decimal point, another series
of numbers, and a null terminator are present.
 *
 * ARGUMENTS:		char		*ptr
		pointer to string to check.
 *
 * RETURNS:		BOOL		FALSE		if string is not a
valid non-negative decimal value
 *
		TRUE		if string is OK
 */

BOOL IsDecimal(char *ptr)
{
		char *dotptr;
		BOOL  bValid;

		if ( *ptr == 0 )
			return FALSE;

		// find decimal point
		dotptr = strchr( ptr, '.' );
		if (dotptr == NULL)
			// no decimal point, so just check for
numeric
			return IsNumeric(ptr);
		*dotptr = 0;  // temporarily replace decimal with a
terminator

		if ( *ptr != 0 )
			bValid = IsNumeric(ptr);
		// string starts with decimal point
		else if (*(dotptr+1) == 0)
			return FALSE;  // nothing but a decimal point
is bad
		else
			bValid = TRUE;

		if (*(dotptr+1) != 0)
			// check text after decimal point
			bValid &= IsNumeric(dotptr+1);

		*dotptr = '.';  // replace decimal point
```

```
        return bValid;
}
```

Isapi_dll/src/resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1                     101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1000
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

common/src/ReadRegistry.h

```
/*      FILE:                   ReadRegistry.h
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      not audited
 *
 *      PURPOSE:    Header for registry related code.
 *
 *  Change history:
 *                  4.20.000 - first version
 */

enum DBPROTOCOL { Unspecified, ODBC, DBLIB };
const char *szDBNames[] = { "Unspecified", "ODBC", "DBLIB" };

enum TXNMON { None, TUXEDO, ENCINA, COM };
const char *szTxnMonNames[] = { "NONE", "TUXEDO", "ENCINA",
"COM" };

//This structure defines the data necessary to keep distinct for each
terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
        enum DBPROTOCOL eDB_Protocol;
        enum TXNMON eTxnMon;
        BOOL bCOM_SinglePool;
        DWORD dwMaxConnections;
        DWORD dwMaxPendingDeliveries;
        DWORD dwNumberOfDeliveryThreads;
        char szPath[128];
        char szDbServer[32];
        char szDbName[32];
        char szDbUser[32];
        char szDbPassword[32];
        wchar_t szSPPrefix[32];     //tpcc_odbc.dll stored
procedures prefix
        DWORD dwConnectDelay; // delay in ms to use in pacing
connection open and close
        BOOL bCallNoDuplicatesNewOrder;     // whether to
check for non-duplicate item ids and call a different New Order SP
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

);
```

ReadRegistry.cpp

```
/*      FILE:                   READREGISTRY.CPP
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      not yet audited
 *
 *      PURPOSE:    Implementation for TPC-C Tuxedo class.
 *      Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.20.000 - first version
 */

/* FUNCTION: ReadTPCCRegistrySettings
 *
 * PURPOSE: This function reads the NT registry for startup parameters.
There parameters are
 *                                      under the TPCC key.
 *
 * RETURNS  FALSE = no errors
 *                                      TRUE  = error reading registry
 */
BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg )
{
        HKEY        hKey;
        DWORD       size;
        DWORD       type;
        DWORD       dwTmp;
        char        szTmp[256];

        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, KEY_READ, &hKey) !=
ERROR_SUCCESS )
                return TRUE;

        // determine database protocol to use; may be either
ODBC or DBLIB
        pReg->eDB_Protocol = Unspecified;
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey, "DB_Protocol", 0, &type,
(BYTE *)&szTmp, &size) == ERROR_SUCCESS )
        {
                if ( !stricmp(szTmp, szDBNames[ODBC]) )
                        pReg->eDB_Protocol = ODBC;
                else if ( !stricmp(szTmp,
szDBNames[DBLIB]) )
                        pReg->eDB_Protocol = DBLIB;
        }

        pReg->eTxnMon = None;
        // determine txn monitor to use; may be either TUXEDO,
or blank
        size = sizeof(szTmp);
        if ( RegQueryValueEx(hKey, "TxnMonitor", 0, &type,
(BYTE *)&szTmp, &size) == ERROR_SUCCESS )
        {
                if ( !stricmp(szTmp,
szTxnMonNames[TUXEDO]) )
                        pReg->eTxnMon = TUXEDO;
                else if ( !stricmp(szTmp,
szTxnMonNames[ENCINA]) )
                        pReg->eTxnMon = ENCINA;
                else if ( !stricmp(szTmp,
szTxnMonNames[COM]) )
                        pReg->eTxnMon = COM;
        }
```

```
	pReg->bCOM_SinglePool = FALSE;
	size = sizeof(szTmp);
	if ( RegQueryValueEx(hKey, "COM_SinglePool", 0, &type,
(BYTE *)&szTmp, &size) == ERROR_SUCCESS )
	{
		if ( !stricmp(szTmp, "YES") )
			pReg->bCOM_SinglePool =
TRUE;
	}

	pReg->dwMaxConnections = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "MaxConnections", 0, &type,
(LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwMaxConnections = dwTmp;

	pReg->dwMaxPendingDeliveries = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "MaxPendingDeliveries", 0,
&type, (LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwMaxPendingDeliveries = dwTmp;

	pReg->dwNumberOfDeliveryThreads = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "NumberOfDeliveryThreads",
0, &type, (LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwNumberOfDeliveryThreads =
dwTmp;

	size = sizeof( pReg->szPath );
	if ( RegQueryValueEx(hKey, "Path", 0, &type, (BYTE
*)&pReg->szPath, &size) != ERROR_SUCCESS )
		pReg->szPath[0] = 0;

	size = sizeof( pReg->szDbServer );
	if ( RegQueryValueEx(hKey, "DbServer", 0, &type, (BYTE
*)&pReg->szDbServer, &size) != ERROR_SUCCESS )
		pReg->szDbServer[0] = 0;

	size = sizeof( pReg->szDbName );
	if ( RegQueryValueEx(hKey, "DbName", 0, &type, (BYTE
*)&pReg->szDbName, &size) != ERROR_SUCCESS )
		pReg->szDbName[0] = 0;

	size = sizeof( pReg->szDbUser );
	if ( RegQueryValueEx(hKey, "DbUser", 0, &type, (BYTE
*)&pReg->szDbUser, &size) != ERROR_SUCCESS )
		pReg->szDbUser[0] = 0;

	size = sizeof( pReg->szDbPassword );
	if ( RegQueryValueEx(hKey, "DbPassword", 0, &type,
(BYTE *)&pReg->szDbPassword, &size) != ERROR_SUCCESS )
		pReg->szDbPassword[0] = 0;

	size = sizeof( pReg->szSPPrefix );
	if ( RegQueryValueExW(hKey, L"SPPrefix", 0, &type,
(BYTE *)&pReg->szSPPrefix, &size) != ERROR_SUCCESS )
		pReg->szSPPrefix[0] = L'\0';

	pReg->dwConnectDelay = 0;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey, "ConnectDelay", 0, &type,
(LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->dwConnectDelay = dwTmp;

	pReg->bCallNoDuplicatesNewOrder = FALSE;
	size = sizeof(dwTmp);
	if ( ( RegQueryValueEx(hKey,
"CallNoDuplicatesNewOrder", 0, &type, (LPBYTE)&dwTmp, &size) ==
ERROR_SUCCESS )
		&& (type == REG_DWORD) )
		pReg->bCallNoDuplicatesNewOrder =
dwTmp;

	RegCloseKey(hKey);

	return FALSE;
}
```

common\src\error.h

```
/*	FILE:		ERROR.H
 *							Microsoft TPC-C
Kit Ver. 4.20.000
 *							Copyright
Microsoft, 1999
 *	All Rights Reserved
 *
 *							Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *	PURPOSE:	Header file for error exception classes.
 *
 *  Change history:
 *                              4.20.000 - updated rev number to match kit
 *                              4.21.000 - fixed bug: CBaseErr needed to
be declared virtual
 */

#pragma once

#ifndef _INC_STRING
	#include <string.h>
#endif

const int m_szMsg_size = 512;
const int m_szApp_size = 64;
const int m_szLoc_size = 64;

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
	int		iError;
			//error id of message
	char		szMsg[256];
	//message to sent to browser
} SERRORMSG;

typedef enum _ErrorLevel
{
	ERR_FATAL_LEVEL		= 1,
	ERR_WARNING_LEVEL	= 2,
	ERR_INFORMATION_LEVEL	= 3
} ErrorLevel;


#define ERR_TYPE_LOGIC
		-1
	//logic error in program; internal error
#define ERR_SUCCESS
		0
	//success (a non-error error)
#define ERR_BAD_ITEM_ID
		1
	//expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST
		2		//expected
delivery post failed
#define	ERR_TYPE_WEBDLL
		3
	//tpcc web generated error
#define	ERR_TYPE_SQL
		4
	//sql server generated error
#define	ERR_TYPE_DBLIB
		5
	//dblib generated error
```

```
#define       ERR_TYPE_ODBC
                                                6
            //odbc generated error
#define ERR_TYPE_SOCKET
                                                7
            //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK
                                                8
            //dblib and odbc only deadlock condition
#define ERR_TYPE_COM
                                                9
            //error from COM call
#define ERR_TYPE_TUXEDO
                                                10
            //tuxedo error
#define ERR_TYPE_OS
                                                11
            //operating system error
#define ERR_TYPE_MEMORY
                                                12
            //memory allocation error
#define ERR_TYPE_TPCC_ODBC
                                                13
            //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB
                                                14
            //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV
                                                15
            //delivery server error
#define ERR_TYPE_TXNLOG
                                                16
            //txn log error
#define ERR_TYPE_BCCONN
                                                17
            //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN
                                                18
            //Benchcraft connection class
#define ERR_TYPE_ENCINA
                                                19
            //Encina error
#define ERR_TYPE_COMPONENT
                                                20
            //error from COM component
#define ERR_TYPE_RTE
                                                21
            //Benchcraft rte
#define ERR_TYPE_AUTOMATION
                                                22
            //Benchcraft automation errors
#define ERR_TYPE_DRIVER
                                                23
            //Driver engine errors
#define ERR_TYPE_RTE_BASE
                                                24
            //Framework errors
#define       ERR_BUF_OVERFLOW
                                                25
            //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP
                                                26
            //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
                                                27
            //OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB
                                                28
            //error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
                                                50
            //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
                                                51
            //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER
                                                52
            //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
                                    53
#define ERR_TYPE_TPCW_ENG_OS
                                    54
#define ERR_TYPE_HTML_RESP
                                                55
#define ERR_TYPE_TPCW_ODBC
                                                56
#define ERR_TYPE_SCHANNEL
                                                57
#define ERR_TYPE_THINK_LIST
                                                58
//------ end TPC-W ----------
#define ERR_TYPE_XML_PROFILE
                                    59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN
                                                60
            //TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE
                                                61
            //TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE
                                    62                  //Tpce Driver
engine errors
#define ERR_TYPE_TPCE_ENG_OS
                                    63                  //Tpce Driver
engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE
                                    64                  //Tpce MEE
Driver engine errors
//#define ERR_TYPE_TPCE_MEE_ENG_OS
                                    65                  //Tpce MEE
Driver engine system errors

#define ERR_INS_MEMORY
            "Insufficient Memory to continue."
#define ERR_UNKNOWN
            "Unknown error."
#define ERR_MSG_BUF_SIZE                    512
#define INV_ERROR_CODE                                  -1
#define ERR_INS_BUF_OVERFLOW    "Insufficient Buffer size to receive
HTML pages."



class CBaseErr
{
public:
            enum Action
            {
                        eNone = 0
            };

            CBaseErr(LPCTSTR szLoc = NULL)
            {
                        m_idMsg             = GetLastError();
            //take the error code immediatelly before it is reset by
other functions


                        if (szLoc)
                        {
                                    m_szLoc = new
char[strlen(szLoc)+1/*m_szLoc_size*/];
                                    strcpy(m_szLoc, szLoc);
                        }
                        else
                                    m_szLoc       = NULL;

                        m_szApp             = new
char[m_szApp_size];

            GetModuleFileName(GetModuleHandle(NULL), m_szApp,
m_szApp_size);
```

```
	}



	CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
	{
		m_idMsg			= idMsg;


		if (szLoc)
		{
			m_szLoc = new
char[strlen(szLoc)+1/*m_szLoc_size*/];
			strcpy(m_szLoc, szLoc);
		}
		else
			m_szLoc	= NULL;

		m_szApp			= new
char[m_szApp_size];

	GetModuleFileName(GetModuleHandle(NULL), m_szApp,
m_szApp_size);
	}

	virtual ~CBaseErr(void)
	{
		if (m_szApp)
			delete [] m_szApp;
		if (m_szLoc)
			delete [] m_szLoc;
	};

	virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
	{
		int			j = 0;
		char		szTmp[512];

		if (szStr)
			j = wsprintf(szTmp,
"%s\n",szStr);
		if (ErrorNum() != INV_ERROR_CODE)
			j += wsprintf(szTmp+j, "Error
= %d\n", ErrorNum());
		if (m_szLoc)
			j += wsprintf(szTmp+j,
"Location = %s\n", GetLocation());

		j += wsprintf(szTmp+j, "%s\n",
ErrorText());

			MessageBox(hwnd, szTmp, m_szApp,
MB_OK);	}

	char *GetApp(void) { return m_szApp; }
	char *GetLocation(void) { return m_szLoc; }
	virtual int ErrorNum() { return m_idMsg; }



	virtual int ErrorType() = 0;// a value which distinguishes
the kind of error that occurred
	virtual char *ErrorTypeStr() = 0; // text representation of
the error type
	virtual char *ErrorText() = 0; // a string (i.e., human
readable) representation of the error
	virtual int ErrorAction() { return eNone; }		//
the function call that caused the error


protected:
	char		*m_szApp;
	char		*m_szLoc;	// code location where the
error occurred
	int			m_idMsg;

	//short		m_errType;
};

class CSocketErr : public CBaseErr
{
public:
	enum Action
	{
		eNone = 0,
		eSend,
		eSocket,
		eBind,
		eConnect,
		eListen,
		eHost,
		eRecv,
		eGetHostByName,
		eWSACreateEvent,
		eWSASend,
		eWSAGetOverlappedResult,
		eWSARecv,
		eWSAWaitForMultipleEvents,
		eWSAStartup,
		eWSAResetEvent,
		eWSAEnumNetworkEvents,
		eWSAEventSelect,
		eSelect,
		eAccept,
	};		eNonRetryable

	CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

	~CSocketErr()
	{
		if (m_szErrorText != NULL)
			delete [] m_szErrorText;
	};

	Action		m_eAction;
	char		*m_szErrorText;

	int			ErrorType() { return
ERR_TYPE_SOCKET;};
	char*		ErrorTypeStr() { return "SOCKET"; }
	char*		ErrorText(void);
	int			ErrorAction() { return
(int)m_eAction; }

};

class CSystemErr : public CBaseErr
{
public:
	enum Action
	{
		eNone = 0,
		eTransactNamedPipe,
		eWaitNamedPipe,
		eSetNamedPipeHandleState,
		eCreateFile,
		eCreateProcess,
		eCallNamedPipe,
		eCreateEvent,
		eCreateThread,
		eVirtualAlloc,
		eReadFile = 10,
		eWriteFile,
		eMapViewOfFile,
		eCreateFileMapping,
		eInitializeSecurityDescriptor,
		eSetSecurityDescriptorDacl,
		eCreateNamedPipe,
		eConnectNamedPipe,
		eWaitForSingleObject,
		eRegOpenKeyEx,
		eRegQueryValueEx = 20,
		ebeginthread,
		eRegEnumValue,
		eRegSetValueEx,
		eRegCreateKeyEx,
		eWaitForMultipleObjects,
```

```
                    eRegisterClassEx,
                    eCreateWindow,
                    eCreateSemaphore,
                    eReleaseSemaphore,
                    eFSeek,
                    eFRead,
                    eFWrite,
                    eTmpFile,
                    eSetFilePointer,
                    eNew,
                    eCloseHandle,
                    eGetOverlappedResult
        };

                                CSystemErr(Action eAction,
LPCTSTR szLocation);
                                CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
        int                     ErrorType() { return
ERR_TYPE_OS;};
        char*       ErrorTypeStr() { return "SYSTEM"; }
        char        *ErrorText(void);
        int                     ErrorAction() { return
(int)m_eAction; }
        void        Draw(HWND hwnd, LPCTSTR szStr = NULL);

        Action      m_eAction;

private:
        char m_szMsg[ERR_MSG_BUF_SIZE];
};

class CMemoryErr : public CBaseErr
{
public:
        CMemoryErr();

        int                     ErrorType() {return
ERR_TYPE_MEMORY;}
        char*       ErrorTypeStr() { return "OUT OF MEMORY";
}
        char*       ErrorText() {return ERR_INS_MEMORY;}
};


class CBufferOverflowErr : public CBaseErr
{
        public:
        CBufferOverflowErr(int,LPTSTR);

        int                     ErrorType() {return
ERR_BUF_OVERFLOW;}
        char*       ErrorTypeStr() { return "BUFFER
OVERFLOW";}char*    ErrorText() {return
ERR_INS_BUF_OVERFLOW;}
};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{
        public:
                enum Action
                {
                        LoadProfile = 1,
                        LoadSchema,
                        ValidateProfile,
                        SaveProfile,
                        LoadFromXML,
                        SaveToXML,
                        ApplyProcessingInstruction,
                        ApplyAttribute,
                        ApplyNode
                };

                CXMLProfileErr(Action eAction, int eCode,
LPCTSTR szLocation)
                {
                        m_eAction = eAction;
                        m_eCode = eCode;
                        m_bOverload = true;
                };
                CXMLProfileErr(Action eAction, int eCode,
LPCTSTR szLocation, char * szMsg)
                {
                        m_eAction = eAction;
                        m_eCode = eCode;
                        strcpy(m_szMsg, szMsg);
                        m_bOverload = false;
                };

                virtual int             ErrorType() {
return ERR_TYPE_XML_PROFILE;};
                virtual char    *ErrorTypeStr() { return "XML
PROFILE"; };
                virtual char    *ErrorText();
                virtual int             ErrorCode() {
return m_eCode; };
                int
        ErrorAction() { return (int)m_eAction; }
                //virtual void Draw(HWND hwnd, LPCTSTR
szStr = NULL)
                //{
                //      ::MessageBox(hwnd, szStr,
m_szLoc, MB_OK);
                //};

        private:        char
        m_szMsg[ERR_MSG_BUF_SIZE];
                        LPCTSTR m_szLoc;
                        int                     m_eCode;
                        bool        m_bOverload;
                        Action      m_eAction;

};
```

common\src\trans.h

```
/*      FILE:                   ERROR.H
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *      PURPOSE:    Header file for error exception classes.
 *
 *  Change history:
 *                  4.20.000 - updated rev number to match kit
 *                  4.21.000 - fixed bug: ~CBaseErr needed to
be declared virtual
 */

#pragma once

#ifndef _INC_STRING
        #include <string.h>
#endif

const int m_szMsg_size = 512;
const int m_szApp_size = 64;
const int m_szLoc_size = 64;

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
        int                     iError;
                //error id of message
        char        szMsg[256];
        //message to sent to browser
```

```
} SERRORMSG;

typedef enum _ErrorLevel
{
        ERR_FATAL_LEVEL                                 = 1,
        ERR_WARNING_LEVEL               = 2,
        ERR_INFORMATION_LEVEL           = 3
} ErrorLevel;


#define ERR_TYPE_LOGIC
                                -1
        //logic error in program; internal error
#define ERR_SUCCESS
                                0
        //success (a non-error error)
#define ERR_BAD_ITEM_ID
                                1
        //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST
                        2               //expected
delivery post failed
#define     ERR_TYPE_WEBDLL
                                3
        //tpcc web generated error
#define     ERR_TYPE_SQL
                                4
        //sql server generated error
#define     ERR_TYPE_DBLIB
                                5
        //dblib generated error
#define     ERR_TYPE_ODBC
                                6
        //odbc generated error
#define ERR_TYPE_SOCKET
                                7
        //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK
                                8
        //dblib and odbc only deadlock condition
#define ERR_TYPE_COM
                                9
        //error from COM call
#define ERR_TYPE_TUXEDO
                                10
        //tuxedo error
#define ERR_TYPE_OS
                                11
        //operating system error
#define ERR_TYPE_MEMORY
                                12
        //memory allocation error
#define ERR_TYPE_TPCC_ODBC
                                13
        //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB
                                14
        //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV
                                15
        //delivery server error
#define ERR_TYPE_TXNLOG
                                16
        //txn log error
#define ERR_TYPE_BCCONN
                                17
        //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN
                                18
        //Benchcraft connection class
#define ERR_TYPE_ENCINA
                                19
        //Encina error
#define ERR_TYPE_COMPONENT
                                20
        //error from COM component
#define ERR_TYPE_RTE
                                21
        //Benchcraft rte
#define ERR_TYPE_AUTOMATION
                                22
        //Benchcraft automation errors
#define ERR_TYPE_DRIVER
                                23
        //Driver engine errors
#define ERR_TYPE_RTE_BASE
                                24
        //Framework errors
#define     ERR_BUF_OVERFLOW
                                25
        //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP
                                26
        //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
                                27
        //OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB
                                28
        //error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
                                50
        //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
                                51
        //error from Tpcwhtml dll
#define ERR_TYPE_TPCW_USER
                                52
        //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
                                53
#define ERR_TYPE_TPCW_ENG_OS
                                54
#define ERR_TYPE_HTML_RESP
                                55
#define ERR_TYPE_TPCW_ODBC
                                56
#define ERR_TYPE_SCHANNEL
                                57
#define ERR_TYPE_THINK_LIST
                                58
//------ end TPC-W ----------
#define ERR_TYPE_XML_PROFILE
                                59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN
                                60
        //TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE
                                61
        //TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE
                                62              //Tpce Driver
engine errors
#define ERR_TYPE_TPCE_ENG_OS
                                63              //Tpce Driver
engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE
                                64              //Tpce MEE
Driver engine errors
//#define ERR_TYPE_TPCE_MEE_ENG_OS
                                65              //Tpce MEE
Driver engine system errors

#define ERR_INS_MEMORY
        "Insufficient Memory to continue."
#define ERR_UNKNOWN
        "Unknown error."
#define ERR_MSG_BUF_SIZE                512
#define INV_ERROR_CODE                                  -1
#define ERR_INS_BUF_OVERFLOW    "Insufficient Buffer size to receive
HTML pages."


class CBaseErr
```

```
{
public:
	enum Action
	{
		eNone = 0
	};

	CBaseErr(LPCTSTR szLoc = NULL)
	{
		m_idMsg			= GetLastError();
	//take the error code immediatelly before it is reset by
other functions


		if (szLoc)
		{
			m_szLoc = new
char[strlen(szLoc)+1/*m_szLoc_size*/];
			strcpy(m_szLoc, szLoc);
		}
		else
			m_szLoc	= NULL;

		m_szApp			= new
char[m_szApp_size];

	GetModuleFileName(GetModuleHandle(NULL), m_szApp,
m_szApp_size);
	}



	CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
	{
		m_idMsg			= idMsg;


		if (szLoc)
		{
			m_szLoc = new
char[strlen(szLoc)+1/*m_szLoc_size*/];
			strcpy(m_szLoc, szLoc);
		}
		else
			m_szLoc	= NULL;

		m_szApp			= new
char[m_szApp_size];

	GetModuleFileName(GetModuleHandle(NULL), m_szApp,
m_szApp_size);
	}
	virtual ~CBaseErr(void)
	{
		if (m_szApp)
			delete [] m_szApp;
		if (m_szLoc)
			delete [] m_szLoc;
	};

	virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
	{
		int			j = 0;
		char	szTmp[512];

		if (szStr)
			j = wsprintf(szTmp,
"%s\n",szStr);
		if (ErrorNum() != INV_ERROR_CODE)
			j += wsprintf(szTmp+j, "Error
= %d\n", ErrorNum());
		if (m_szLoc)
			j += wsprintf(szTmp+j,
"Location = %s\n", GetLocation());

		j += wsprintf(szTmp+j, "%s\n",
ErrorText());

		MessageBox(hwnd, szTmp, m_szApp,
MB_OK);
	}

	char *GetApp(void) { return m_szApp; }
	char *GetLocation(void) { return m_szLoc; }
	virtual int ErrorNum() { return m_idMsg; }



	virtual int ErrorType() = 0;// a value which distinguishes
the kind of error that occurred
	virtual char *ErrorTypeStr() = 0; // text representation of
the error type
	virtual char *ErrorText() = 0; // a string (i.e., human
readable) representation of the error
	virtual int ErrorAction() { return eNone; }		//
the function call that caused the error


protected:
	char	*m_szApp;
	char	*m_szLoc;	// code location where the
error occurred
	int			m_idMsg;
};		//short	m_errType;

class CSocketErr : public CBaseErr
{
public:
	enum Action
	{
		eNone = 0,
		eSend,
		eSocket,
		eBind,
		eConnect,
		eListen,
		eHost,
		eRecv,
		eGetHostByName,
		eWSACreateEvent,
		eWSASend,
		eWSAGetOverlappedResult,
		eWSARecv,
		eWSAWaitForMultipleEvents,
		eWSAStartup,
		eWSAResetEvent,
		eWSAEnumNetworkEvents,
		eWSAEventSelect,
		eSelect,
		eAccept,
		eNonRetryable
	};

	CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

	~CSocketErr()
	{
		if (m_szErrorText != NULL)
			delete [] m_szErrorText;
	};

	Action	m_eAction;
	char	*m_szErrorText;

	int			ErrorType() { return
ERR_TYPE_SOCKET;};
	char*	ErrorTypeStr() { return "SOCKET"; }
	char*	ErrorText(void);
	int			ErrorAction() { return
(int)m_eAction; }

};

class CSystemErr : public CBaseErr
{
public:
```

```
        enum Action
        {
                eNone = 0,
                eTransactNamedPipe,
                eWaitNamedPipe,
                eSetNamedPipeHandleState,
                eCreateFile,
                eCreateProcess,
                eCallNamedPipe,
                eCreateEvent,
                eCreateThread,
                eVirtualAlloc,
                eReadFile = 10,
                eWriteFile,
                eMapViewOfFile,
                eCreateFileMapping,
                eInitializeSecurityDescriptor,
                eSetSecurityDescriptorDacl,
                eCreateNamedPipe,
                eConnectNamedPipe,
                eWaitForSingleObject,
                eRegOpenKeyEx,
                eRegQueryValueEx = 20,
                ebeginthread,
                eRegEnumValue,
                eRegSetValueEx,
                eRegCreateKeyEx,
                eWaitForMultipleObjects,
                eRegisterClassEx,
                eCreateWindow,
                eCreateSemaphore,
                eReleaseSemaphore,
                eFSeek,
                eFRead,
                eFWrite,
                eTmpFile,
                eSetFilePointer,
                eNew,
                eCloseHandle,
                eGetOverlappedResult
        };

                        CSystemErr(Action eAction,
LPCTSTR szLocation);
                        CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
        int             ErrorType() { return
ERR_TYPE_OS;};
        char*   ErrorTypeStr() { return "SYSTEM"; }
        char    *ErrorText(void);
        int             ErrorAction() { return
(int)m_eAction; }
        void    Draw(HWND hwnd, LPCTSTR szStr = NULL);
        Action  m_eAction;

private:
        char m_szMsg[ERR_MSG_BUF_SIZE];
};

class CMemoryErr : public CBaseErr
{
public:
        CMemoryErr();

        int             ErrorType() {return
ERR_TYPE_MEMORY;}
        char*   ErrorTypeStr() { return "OUT OF MEMORY";
}
        char*   ErrorText() {return ERR_INS_MEMORY;}
};


class CBufferOverflowErr : public CBaseErr
{
        public:
        CBufferOverflowErr(int,LPTSTR);

        int             ErrorType() {return
ERR_BUF_OVERFLOW;}
        char*   ErrorTypeStr() { return "BUFFER
OVERFLOW"; }
        char*   ErrorText() {return
ERR_INS_BUF_OVERFLOW;}
};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{
        public:
                enum Action
                {
                        LoadProfile = 1,
                        LoadSchema,
                        ValidateProfile,
                        SaveProfile,
                        LoadFromXML,
                        SaveToXML,
                        ApplyProcessingInstruction,
                        ApplyAttribute,
                        ApplyNode
                };

                CXMLProfileErr(Action eAction, int eCode,
LPCTSTR szLocation)
                {
                        m_eAction = eAction;
                        m_eCode = eCode;
                        m_bOverload = true;
                };
                CXMLProfileErr(Action eAction, int eCode,
LPCTSTR szLocation, char * szMsg)
                {
                        m_eAction = eAction;
                        m_eCode = eCode;
                        strcpy(m_szMsg, szMsg);
                        m_bOverload = false;
                };

                virtual int             ErrorType() {
return ERR_TYPE_XML_PROFILE;};
                virtual char    *ErrorTypeStr() { return "XML
PROFILE"; };
                virtual char    *ErrorText();
                virtual int             ErrorCode() {
return m_eCode; };
                int
        ErrorAction() { return (int)m_eAction; }
                //virtual void Draw(HWND hwnd, LPCTSTR
szStr = NULL)
                //{
                //              ::MessageBox(hwnd, szStr,
m_szLoc, MB_OK);
                //};
        private:
                char
        m_szMsg[ERR_MSG_BUF_SIZE];
                LPCTSTR m_szLoc;
                int             m_eCode;
                bool    m_bOverload;
                Action  m_eAction;

};
```

Common\src\txn_base.h

```
/*      FILE:           TXN_BASE.H
 *                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                      Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
```

```
 *
 *          PURPOSE:    Header file for TPC-C txn class
implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */

#pragma once

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
        public:
                CTPCC_BASE(void) {};
                virtual ~CTPCC_BASE(void) {};

                virtual PNEW_ORDER_DATA
        BuffAddr_NewOrder()                 = 0;
                virtual PPAYMENT_DATA
        BuffAddr_Payment()                  = 0;
                virtual PDELIVERY_DATA
        BuffAddr_Delivery()                 = 0;
                virtual PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()       = 0;
                virtual PORDER_STATUS_DATA
        BuffAddr_OrderStatus()      = 0;

                virtual void NewOrder                   () =
0;
                virtual void Payment                    () =
0;
                virtual void Delivery                   () =
0;
                virtual void StockLevel                 () =
0;
                virtual void OrderStatus    () = 0;
};
```

\common\txnlog\include\rtetime.h

```
/* FILE: rtetime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Source code licensed to Tandem Computers  for Internal
 * use only.  Redistribution of source or object files or
 * any derivative works is prohibited. By agreement, this
 * notice may not be removed.
 *
 * Authors: Charles Levine, Philip Durr
 *                                    Microsoft Corp.
 *
 */

//FILE: RTETIME.H

#define     MAX_JULIAN_TIME
            0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME     __int64
#define     TC_TIME                 DWORD
extern "C"
{
BOOL                InitJulianTime(LPSYSTEMTIME lpInitTime);
JULIAN_TIME         GetJulianTime(void);
DWORD               MyTickCount(void);
void                GetJulianAndTC(JULIAN_TIME *pJulian,
DWORD *pTC);
JULIAN_TIME         ConvertTo64BitTime(int iYear, int iMonth, int
iDay, int iHour, int iMinute, int iSecond);
JULIAN_TIME         Get64BitTime(LPSYSTEMTIME lpInitTime);
int                             JulianDay( int yr, int mm, int
dd );
void                JulianToTime(JULIAN_TIME julianTS, int* yr,
int* mm, int* dd, int *hh, int *mi, int *ss );
void                JulianToCalendar( int day, int* yr, int* mm,
int* dd );
}
```

\common\txnlog\include\spinlock.h

```
/*          FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Source code licensed to Tandem Computers  for Internal
 * use only.  Redistribution of source or object files or
 * any derivative works is prohibited. By agreement, this
 * notice may not be removed.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 */

#ifndef _INC_Spinlock

        const LONG LockClosed     = 1;
        const LONG LockOpen                 = 0;

        /*********************************************
*****************
         *
         *  Spinlock and Semaphore locking.
         *
         *  This class provides a very conservative locking
scheme.
         *  The assumption behind the code is that locks will be
         *  held for a very short time.  When a lock is taken a
memory
         *  location is exchanged.  All other threads that want this
         *  lock wait by spinning and sometimes sleeping on a
semaphore
         *  until it becomes free again.  The only other choice is
not
         *  to wait at all and move on to do something else.  This
         *  module should normally be used in conjunction with
cache
         *  aligned memory in minimize cache line misses.
         *

*********************************************************
*****/

        class Spinlock
        {
                // Private data.
                HANDLE              Semaphore;
                volatile LONG       m_Spinlock;
                volatile LONG       Waiting;

                #ifdef _DEBUG
                        // Counters for debugging
builds.
                        volatile LONGTotalLocks;
                        volatile LONGTotalSleeps;
                        volatile LONGTotalSpins;
                        volatile LONGTotalWaits;
                #endif

                public:
                        // Public functions.

                        Spinlock( void );
```

```
                                inline BOOL ClaimLock( BOOL
Wait = TRUE );
                                inline void ReleaseLock( void );
                                ~Spinlock( void );
                                // Disabled operations.
                                Spinlock( const Spinlock &
Copy );
                                void operator=( const Spinlock
& Copy );

                        private:
                                // Private functions.
                                inline BOOL ClaimSpinlock(
volatile LONG *sl );
                                void WaitForLock( void );
                                void WakeAllSleepers( void );
        };

        /*****************************************************
****************
         *
         *  A guaranteed atomic exchange.
         *
         *  An attempt is made to claim the Spinlock.  This action
is
         *  guaranteed to be atomic.
         *
*************************************************************
****/

        inline BOOL Spinlock::ClaimSpinlock( volatile LONG
*Spinlock )
        {
                #ifdef _DEBUG
                        InterlockedIncrement(
(LPLONG) & TotalLocks );
                #endif
                return ( ((*Spinlock) == LockOpen) &&
(InterlockedExchange( (LPLONG)Spinlock, LockClosed ) == LockOpen)
);
        }

        /*****************************************************
*****
         *
         *  Claim the Spinlock.
         *
         *  Claim the lock if available else wait or exit.
         *
*****************************************************/

        inline BOOL Spinlock::ClaimLock( BOOL Wait )
        {
                if ( ! ClaimSpinlock( (volatile LONG*) &
m_Spinlock ) )
                {
                        if ( Wait )
                                WaitForLock();
                        return Wait;
                }
                return TRUE;
        }

        /*****************************************************
**********
         *
         *  Release the Spinlock.
         *
         *  Release the lock and if needed wakeup any sleepers.
         *

*****************************************************/

        inline void Spinlock::ReleaseLock( void )
        {
                m_Spinlock = LockOpen;
                if ( Waiting > 0 )
                        WakeAllSleepers();
        }

        #define _INC_Spinlock

#endif
```

\common\txnlog\include\txnlog.h

```
/*      FILE:           TXNLOG.H
 *                                      Microsoft TPC-C
Kit Ver. 4.10.000
 *                                      not yet audited
 *
 *      PURPOSE:    Header file for txn log class
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 */
#include <stdio.h>              //needed for FILE

#define DRIVER_NAME_LEN
                        32      //max length of the driver
engine name - must be the same as in engstut.h!
#define TXN_LOG_INCORRECTLY_SHUT_DOWN   100
        //ctrl rec subtype generated by the txn log when reading
an abruptly shut down log

#pragma once

   typedef struct _TXN_NEWORDER
        {
     BYTE   OL_Count;        //range 0 to 31
     BYTE   OL_Remote_Count;    //range 0 to 31
     WORD  c_id;
     int            o_id;
        } TXN_NEWORDER;

   typedef struct _TXN_PAYMENT
        {
     BYTE   CustByName;
     BYTE   IsRemote;
        } TXN_PAYMENT;

   typedef struct _TXN_ORDERSTATUS
        {
                BYTE        CustByName;
        } TXN_ORDERSTATUS;

        typedef union _TXN_DETAILS
        {
                TXN_NEWORDER        NewOrder;
                TXN_PAYMENT
        Payment;
                TXN_ORDERSTATUS     OrderStatus;
        } TXN_DETAILS;


        // Common header for all records in txn log.  The TxnType
field is
        // a switch which identifies the particular variant.
        #define TXN_REC_TYPE_CONTROL            1
        //
        #define TXN_REC_TYPE_TPCC
        2       // replaces TRANSACTION_TYPE_TPCC
        #define TXN_REC_TYPE_TPCC_DELIV_DEF     3

        #define TXN_REC_TYPE_TPCW
        4       // replaces TRANSACTION_TYPE_TPCW

        typedef struct _TXN_RECORD_HEADER
        {
                JULIAN_TIME         TxnStartT0;
        // start of txn
```

```
                BYTE        TxnType;
        // one of TXN_REC_TYPE_*
                BYTE        TxnSubType;
        // depends on TxnType
        } TXN_RECORD_HEADER, *PTXN_RECORD_HEADER;


        typedef struct _TXN_RECORD_CONTROL
        {
                // common header; must exactly match
TXN_RECORD_HEADER
                JULIAN_TIME             TxnStartT0;
        // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_CONTROL
                BYTE        TxnSubType;
        // depends on TxnType
                // end of common header

                DWORD       Len;
                // number of bytes after this field
        } TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;


        // TPC-C Txn Record Layout:
        //
   //'TxnStartT0' is a Julian timestamp corresponding to the moment
the
   //txn is sent to the SUT, i.e., beginning of response time.  Deltas
   //are in milliseconds.  Note that if RTDelay > 0, then the txn was
   //delayed by this amount.  The delay occurs at the beginning of the
   //response time.  So if RTDelay > 0, then the txn was actually sent
   //at TxnStartT0 + RTDelay.
   //
   //Graphically:
   //
   //  time -->
   //
   //    |--- Menu  ---|-- Keying  --|-- Response --|--- Think  --|
   //     <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4  -> <- DeltaT3 ->
   //                   ^
   //                   ^ TxnStartT0
   //
   //RTDelay is the amount of response time delay included in DeltaT4.
   //RTDelay is recorded per txn because this value can be changed on
   //the fly, and so may vary from txn to txn.
   //
   //TxnStatus is the txn completion code.  It is used to indicate errors.
   //For example, in the New Order txn, 1% of txns abort.  TxnStatus
will
   //reflect this.
```

```
        typedef struct _TXN_RECORD_TPCC
                // common header; must exactly match
TXN_RECORD_HEADER
                JULIAN_TIME             TxnStartT0;
        // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_TPCC
                BYTE        TxnSubType;
        // depends on TxnType
                // end of common header

                int     DeltaT1;                    //
menu time (ms)
                int     DeltaT2;                    //
keying time (ms)
                int     DeltaT3;                    //
think time (ms)
                int     DeltaT4;                    //
response time (ms)
                int     RTDelay;                    //
response time delay (ms)
                int                     TxnError;
        // error code providing more detail for TxnStatus
                int                     w_id;
                // warehouse ID
                BYTE        d_id;
        // assigned district ID for this thread
                BYTE        d_id_ThisTxn;           //
district ID chosen for this particular
                BYTE        TxnStatus;              //
completion status for txn to indicate errors
                BYTE        reserved;               //
for word alignment
                TXN_DETAILS             TxnDetails;   //

                bool IsSuccessRecord() { return (TxnStatus
== ERR_SUCCESS || TxnStatus == ERR_BAD_ITEM_ID || TxnStatus
== ERR_TYPE_DELIVERY_POST); }
        } TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;


        // TPC-C Deferred Delivery Txn Record Layout:
        //
   //Incorporating delivery transaction information into the above
        //structure would increase the size of TXN_DETAILS from
8 to 42 bytes.
   //Hence, we store delivery transaction details in a separate structure.
   //
        typedef struct _TXN_RECORD_TPCC_DELIV_DEF
        {
                // common header; must exactly match
TXN_RECORD_HEADER
                JULIAN_TIME             TxnStartT0;
        // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_TPCC_DELIV_DEF;
        // = 0
                BYTE        TxnSubType;
                // end of common header

                int     DeltaT4;                    //
response time (ms)
                int                     DeltaTxnExec;
        // execution time (ms)
                int                     w_id;
                // warehouse ID
                BYTE        TxnStatus;              //
completion status for txn to indicate errors
                BYTE        reserved;               //
for word alignment
                short   o_carrier_id;       // carrier id
                long        o_id[10];               //
returned delivery transaction ids

                bool IsSuccessRecord() { return (TxnStatus
== ERR_SUCCESS || TxnStatus == ERR_BAD_ITEM_ID || TxnStatus
== ERR_TYPE_DELIVERY_POST); }
        } TXN_RECORD_TPCC_DELIV_DEF,
*PTXN_RECORD_TPCC_DELIV_DEF;


        //
        //TPC-W records.
        //
        typedef struct _TXN_RECORD_TPCW
        {
                // common header; must exactly match
TXN_RECORD_HEADER
                JULIAN_TIME             TxnStartT0;
        // start of txn
                BYTE        TxnType;
        // = TXN_REC_TYPE_TPCW
                BYTE        TxnSubType;
        // depends on TxnType
                // end of common header

                int     ThinkTime;                  //
think time (ms)
                int     WIRT;                       //
response time (ms)
                int                     TxnError;
        // error code providing more detail for TxnStatus
                BYTE        TxnStatus;              //
completion status for txn to indicate errors
                //This field below depends on the txn sub
type:
                //- for Home interaction:   it indicates
whether the user was a new customer (or returning)
```

```
                        //- for Buy Confirm:                    it
indicates whether the shipping address was updated
                        //- for Search Request:                 it
indicates the search type (Author, Title, or Subject)
                        //This statistics needs to be reported
according to 5.5.5.1 clause in the specs.
                        //Because this field occupies 1 byte, the
record structure is already aligned on word boundary.
                        union           {
                                BYTE            newCustomer;
                                BYTE            addrUpdated;
                                BYTE            searchType;
                        }               intrDetails;

                        //This field is mostly for
informational/debugging purposes.
                        //It indicates what user performed this web
interaction and what instance (session) of that use it was.
                        //The first 22 bits indicate the user #, and
the top 10 bits indicate instance (session) #.
                        unsigned __int32            uiUser;

                        bool IsSuccessRecord() { return (TxnStatus
== ERR_SUCCESS); }
                } TXN_RECORD_TPCW, *PTXN_RECORD_TPCW;

                //
                //              Data part of a control record written when a
user is created (or it's new session) - to record USMD
                typedef struct _TXN_RECORD_TPCW_USER_DATA
                {
                        unsigned __int32            uiUser;
                                // user number
                        JULIAN_TIME
                USMD;                                                   //
USMD for this user
                        BYTE
                bRetCust;                               // returning
customer?
                } TXN_RECORD_TPCW_USER_DATA,
*PTXN_RECORD_TPCW_USER_DATA;

                //The entire TPCW User control record structure
                typedef struct _TXN_RECORD_TPCW_USER
                {
                        // common header; must exactly match
TXN_RECORD_HEADER
                        JULIAN_TIME                 TxnStartT0;
                // start of txn
                        BYTE            TxnType;
                // = TXN_REC_TYPE_CONTROL
                        BYTE            TxnSubType;
                // depends on TxnType
                        // end of common header

                        DWORD       Len;
                        // number of bytes after this field
                        //The fields above must exactly match
TXN_RECORD_CONTROL

                        //The fields below must exactly match
TXN_RECORD_TPCW_USER_DATA
                        unsigned __int32            uiUser;
                                // user number
                        JULIAN_TIME
                USMD;                                                   //
USMD for this user
                        BYTE
                bRetCust;                               // returning
customer?
                } TXN_RECORD_TPCW_USER,
*PTXN_RECORD_TPCW_USER;


#define                 USER_INDEX_NBITS        22
#define                 USER_INDEX_MASK
                0x003fffff                  //lower 22 bits mask for user
field in TPCW record
#define                 USER_SESSION_MASK       0xffc00000
                //upper 10 bits mask for user field in TPCW record
#define                 USER_CREATE_REC                         254
                        //subtype for the control record written
when a user is created

#define                 TXN_LOG_VERSION                         2
#define                 TXN_DATA_START
                4096    // offset in log file where log records start
#define                 TXN_LOG_EYE_CATCHER  "BC"           //
signature bytes at the start of log file


                ////////////////////////////////////////////////////////////////////
/////////
                // The transaction log has a header as the first 4K block.
                //
                typedef struct _TXN_LOG_HEADER
                {
                        char
                EyeCatcher[2];              // signature bytes; should
always be "BC"
                        int
                LogVersion;                 // set to TXN_LOG_VERSION
                        JULIAN_TIME
                BeginTxnTS;                 // timestamp of first (lowest)
txn start
                        JULIAN_TIME
                EndTxnTS;                   // timestamp of last (highest)
txn completion time
                        int
                iRecCount;                  // number of records in log file
                        BOOL
                bLogSorted;
                        int
                iFileSize;                  // file size in bytes

                        // driver engine that created this log file
                        char
                szDriverEngineName[DRIVER_NAME_LEN];
                        // the record map provides a fast way to get
close to a particular timestamp in a sorted log file.
                //      struct
                //      {
                //              JULIAN_TIME
                TS;                                 // timestamp of
record
                //              int
                                iPos;               // byte position in
file
                //      }
                RecMap[RecMapSize];
//#define                 RecMapSize                              200

                } TXN_LOG_HEADER, *PTXN_LOG_HEADER;

                /* Header of the sorted pointers blocks in Temp file (in
merging). */
                typedef struct BLOCK_HEADER {
                        long            BlockPos;
                        __int64 CurPos;
                        DWORD           BytesRead;
                        int                             nRecords;
                        BYTE            *offset;        /* offset of
pointers to records in the log file */
                } BLOCK_HEADER, *PBLOCK_HEADER;



#define         READ_BUFFER_SIZE                64*1024
//#define       WRITE_BUFFER_SIZE                8*1024
#define         WRITE_BUFFER_SIZE                128*1024

#define         NUM_READ_BUFFERS                1
#define         NUM_WRITE_BUFFERS               2
#define         MAX_NUM_BUFFERS                                 2

// flags passed in to the constructor
#define         TXN_LOG_WRITE                   0x01
#define         TXN_LOG_READ                    0x02
```

```
#define     TXN_LOG_SORTED                  0x04
#define TXN_LOG_CRASHOPEN       0x08                    // if
set, invalid headers will be tolerated; used for recovery

#define     TXN_LOG_OS_ERROR    1
#define     TXN_LOG_NOT_SORTED  2

#define SKIP_CTRL_RECS              1

class CTxnLog
{
        private:
                DWORD               iBufferSize;
                //buffer allocated size
                DWORD
        iBytesFreeInBuffer;                 //total bytes
available for use in buffer
                int
        iNumBuffers;                        //buffers in use
                int
        iActiveBuffer;                      //indicates which
buffer is active: 0 or 1
                int
        iIoBuffer;
        //buffer for any pending IO operation
//              int
        iFilePointer;                       //position in file.
                LARGE_INTEGER       lFilePointer;
        //position in file.
                int
        iNextRec;
        //when reading, ordinal value of next record

                // A "save point" is remembered each time
GetNextRecord is called with a start time specified.
                // The next time it is called, if start time is
after the save point, we start scanning from the
                // save point.  This is particularly useful in
FindBestInterval, where the log is scanned repeatedly.
                JULIAN_TIME
        SavePtTime;
//              int
        iSavePtFilePointer;
                LARGE_INTEGER
        lSavePtFilePointer;
                int
        iSavePtNextRec;

                JULIAN_TIME         lastTS;
                                    //when writing
sorted output, used to verify records are sorted
                BOOL                bWrite;
                BOOL                bCrashOpen;
                                    // writing to file
                        // tolerate bad headers and
consistency checks

                BOOL
        bLogSorted;                 // is log file
sorted?  applies to both input and output
                JULIAN_TIME
        BeginTxnTS;                 // timestamp of
first (lowest) txn start
                JULIAN_TIME
        EndTxnTS;                   // timestamp of
last (highest) txn completion time
                int
        iRecCount;                  // number of
records in log file
                // To write a checkpoint information into the
header, need to know the EndTxnTS for the
                // last record written to the disk. It is not
necessarily the last record in the
                // last written buffer, as the last record may
be only partially in the buffer.
                // So remember the timestamps for 2 last
records that begin in the buffer - one of
                // them will the last complete record written
to disk.
                JULIAN_TIME
        PrevEndTxnTS;               // timestamp of
the previous to last record
                union {
                        TXN_LOG_HEADER
        HeaderForCheckpoint;// header written on every
checkpoint
                        char
                szHeaderBuffer[512];    // 512 bytes is
the minimum we can write to the disk
                } HeaderBuffer;         //need the union
because can't write sizeof(TXN_LOG_HEADER) - too few bytes

                // Control record returned from
GetNextRecord if the file
                // currently opened for read was not
properly shut down
                struct
                {
                        TXN_RECORD_CONTROL
        RecHeader;
                        char
                szDriverName[DRIVER_NAME_LEN];

                } IncorrectShutDownRec;
                BYTE                *pCurrent;
                BYTE        //ptr to current buffer
        *pBuffer[MAX_NUM_BUFFERS];

                PTXN_RECORD_HEADER  *TxnArray;
        //transaction record pointer array for sort

                DWORD               dwError;
                DWORD
        dwCheckpointError;          //error in
checkpoint thread
                HANDLE              hTxnFile;
                        //handle to log file
                HANDLE              hMapFile;
                        //map file used when sorting
the log
                HANDLE              hIoComplete;
                //event to signify that there are no pending
IOs
                HANDLE              hLogFileIo;
                        //event to signal the IO thread
to write the inactive buffer
                HANDLE
        hStopCheckpointThread;  //event to signal the
checkpoint thread to exit

                Spinlock    Spin;
                        //spin lock to protect the txn
log file buffers
                Spinlock    WriteSpin;
                //spin lock to protect the WriteFile operation
between IO and Checkpoint threads

                FILE
        *tmpFile;           //temp file for merging sorted
pieces
                PBLOCK_HEADER
        tmpHeaders;         //sorted pointers block header
                BYTE
        **recPointers;      //record pointer buffers for
each sorted block
                PTXN_RECORD_HEADER  *recBuffers;
        //record buffers for each sorted block
                int                 *PointersRead;
                                    //# of pointers
processed in each block
                BOOL        *BlockAvailable;
                //whether to check a particular block for
jmin

                int                 nBlocks;
```

```
                int                     jmin;

        //index (block-wise) of the lowest timestamp record
                int                     iAvgRecordLen;
                                        //average record
length

                int
        iSortedReturnedCount;           //keeps track of
the # of sorted records returned through GetSortedRecord()

                BOOL        bIncorrectShutDown;
                // indicates whether the log opened for read
was not correctly shut down

                int Write(BYTE *ptr, DWORD Size);
                static void LogFileIO(CTxnLog *);

                void LoadBuffers(int j);
        //used in sort/merge to load record buffers

                static void CheckpointThread(CTxnLog *);
        // checkpointing thread

        public:

                CTxnLog(LPCTSTR szFileName, DWORD
dwOpts, char *szDriver = NULL);
                ~CTxnLog(void);

                int WriteToLog(PTXN_RECORD_TPCC
pTxnRcrd);
                int
WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
                int WriteToLog(PTXN_RECORD_CONTROL
pCtrlRec);
                int WriteToLog(PTXN_RECORD_HEADER
pCtrlRec);
                int WriteToLog(PTXN_RECORD_TPCW
pTxnRcrd);   //support for TPC-W

                int WriteCtrlRecToLog(BYTE SubType,
LPTSTR lpStr, DWORD dwLen);

                void CloseTransactionLogFile(void);

                PTXN_RECORD_HEADER
GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
                PTXN_RECORD_HEADER
GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL bSkipCtrlRecs =
FALSE);
                PTXN_RECORD_HEADER GetSortedRecord();
                int Sort(void);
                inline BOOL IsSorted(void) { return
bLogSorted; };
                inline JULIAN_TIME BeginTS(void) { return
BeginTxnTS; };
                inline JULIAN_TIME EndTS(void) { return
EndTxnTS; };
                inline int RecordCount(void) { return
iRecCount; };
};

class CTXNLOG_ERR : public CBaseErr
{
        public:
                enum CTXNLOG_ERRS
                {
                        ERR_BAD_FILE_FORMAT,
        // "File format is invalid."

        ERR_UNKNOWN_LOG_VERSION,        // "Log file
version is unknown."
                        ERR_BROKEN_LOG_FILE,
        // "Log file is broken."
                        ERR_LOG_NOT_SORTED,
                // "Log file is not sorted"
                        ERR_INVALID_TIME_SEQ,
        // "Internal Error: Record Time Sequence invalid."
                };

                CTXNLOG_ERR(int iErr) : CBaseErr(iErr) {};

                int ErrorType() {return
ERR_TYPE_TXNLOG;};
                char *ErrorTypeStr() { return "TXN LOG"; }

                char *ErrorText()
                {
                        static char *szMsgs[] = {
                                "File format is
invalid.",
                                "Log file version
is unknown.",
                                "Log file is
broken.",
                                "Log file is not
sorted",
                                "Internal Error:
Record Time Sequence invalid.",
                                ""
                        };

                        for(int i = 0; szMsgs[i][0]; i++)
                        {               if ( m_idMsg == i
)
                                break;
                        }

                        return(szMsgs[i][0] ? szMsgs[i]
: ERR_UNKNOWN);
                };
};
```

db_dblib_dll\src\tpcc_dlib.cpp

```
/*              FILE:                   TPCC_DBLIB.CPP
 *                                              Microsoft TPC-C
Kit Ver. 4.42.000
 *                                              Copyright
Microsoft, 2002
 *              All Rights Reserved
 *
 *                                              Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:    Implements dblib calls for TPC-C txns.
 *              Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.42.000 - changed w_id fields from short to
long to support >32K warehouses
 *                      4.20.000 - updated rev number to match kit
 *                      4.10.001 - not deleting error class in catch
handler on deadlock retry;
 *                                                      not a functional
bug, but a memory leak
 *                                                      - had to tweak
some declarations to compile with latest SDK; no functional change
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqlfront.h>
#include <sqldb.h>

#ifdef ICECAP
```

```
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define     DEFCLPACKSIZE
            4096

// version string; must match return value from tpcc_version stored
proc
const char    sVersion[] = "4.20.000";

const                               iMaxRetries = 10;
            // how many retries on deadlock
static      long            iConnectionCount = 0;  // number of current
dblib connections

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
            switch( ul_reason_for_call )
            {
                        case DLL_PROCESS_ATTACH:

            DisableThreadLibraryCalls(hModule);
                                    dbinit();                   //
initialize dblib
                                    break;

                        case DLL_PROCESS_DETACH:
                                    dbexit();                   //
close all dblib structures/connections
                                    break;

                        default:
                                    /* nothing */;
            }
            return TRUE;
}


int err_handler(DBPROCESS *dbproc, int severity, int dberr, int oserr,
LPCSTR dberrstr, LPCSTR oserrstr)
{           CTPCC_DBLIB
            *pConn;

            assert(dbproc != NULL);
            pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

            if (pConn != NULL)
            {
                        pConn->SetDbLibError( severity, dberr,
oserr, dberrstr, oserrstr );
            }
            return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity, char *msgtext)
 *
 * PURPOSE: This function handles DB-Library SQL Server error
messages
 *
 * ARGUMENTS:          DBPROCESS           *dbproc
                       DBPROCESS id pointer
 *                                         DBINT
                       msgno
            message number
 *                                         int
                                   msgstate
            message state
 *                                         int
                                   severity
            message severity
 *                                         char
                       *msgtext            printable
message description
 *
 * RETURNS:            int
            INT_CONTINUE           continue if error is SQLETIME
else INT_CANCEL action
 *
                                   INT_CANCEL
            cancel operation
 *
 * COMMENTS:           This function also sets the dead lock dbproc
variable if necessary.
 *
 */

// typedef INT (SQLAPI *DBMSGHANDLE_PROC)(PDBPROCESS, DBINT,
INT, INT, LPCSTR, LPCSTR, LPCSTR, DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int
severity,
                                                LPCSTR msgtext,
LPCSTR srvname, LPCSTR procname, DBUSMALLINT line)
{           CTPCC_DBLIB
            *pConn;

            assert(dbproc != NULL);
            pConn = (CTPCC_DBLIB*)dbgetuserdata(dbproc);

            if (pConn != NULL)
            {
                        pConn->SetSqlError( msgno, msgstate,
severity, msgtext );
            }

            return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
 *
 * PURPOSE: This function copies n characters from string pSrc to pDst
and places a
 *                                   null character at the end of the
destination string.
 *
 * ARGUMENTS:          char
            *pDest      destination string pointer
 *                                         char
                       *pSrc       source string pointer
 *                                         int
                                   n
            number of characters to copy
 *
 * RETURNS:            None
 *
 * COMMENTS:           Unlike strncpy this function ensures that the
result string is
 *                                         always null
terminated.
 *
 */

inline static void UtilStrCpy(char * pDest, const BYTE * pSrc, int n)
{
   strncpy(pDest, (char *)pSrc, n);
   pDest[n] = '\0';

            return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */
```

```
char* CTPCC_DBLIB_ERR::ErrorText(void)
{
	int i;

	static SERRORMSG errorMsgs[] =
	{
		{ ERR_WRONG_SP_VERSION,
	"Wrong version of stored procs on database server"	},
		{ ERR_INVALID_CUST,
	"Invalid Customer id,name."
									},
		{ ERR_NO_SUCH_ORDER,		"No
orders found for customer."
				},
		{ ERR_RETRIED_TRANS,
	"Retries before transaction succeeded."
		},
		{ 0,
				""



	}
	};

	static char szNotFound[] = "Unknown error number.";

	for(i=0; errorMsgs[i].szMsg[0]; i++)
	{
		if ( m_errno == errorMsgs[i].iError )
			break;
	}
	if ( !errorMsgs[i].szMsg[0] )
		return szNotFound;
	else
		return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
	LPCSTR szServer,			// name of SQL
server
	LPCSTR szUser,					//
user name for login
	LPCSTR szPassword,			// password for
login
	LPCSTR szHost,					//
workstation name; shows up in sp_who; max 30 chars, only first 10
kept by SQL Server
	LPCSTR szDatabase )			// name of
database to use
{
	return new CTPCC_DBLIB( szServer, szUser, szPassword,
szHost, szDatabase );
}


CTPCC_DBLIB::CTPCC_DBLIB (
	LPCSTR szServer,			// name of SQL
server
	LPCSTR szUser,					//
user name for login
	LPCSTR szPassword,			// password for
login
	LPCSTR szHost,					//
workstation name; shows up in sp_who; max 30 chars, only first 10
kept by SQL Server
	LPCSTR szDatabase )			// name of
database to use
{
	LOGINREC    *login;
	const BYTE   *pData;

	// initialization
	m_dbproc = NULL;
	m_DbLibErr = (CDBLIBERR*)NULL;
	m_SqlErr = (CSQLERR*)NULL;

	m_MaxRetries = 10;			// how many
retries on deadlock

	// increase max number of connections if getting close
	if ( dbgetmaxprocs() < (iConnectionCount+5) )
	{
		if ( dbsetmaxprocs(iConnectionCount+10)
== FAIL )

	ThrowError(CDBLIBERR::eDbSetMaxProcs);
	}

	// allocate a login structure
	login = dblogin();
	if (login == NULL)
		ThrowError(CDBLIBERR::eLogin);
	InterlockedIncrement( &iConnectionCount );

	// register error and message handler functions
	if (dbprocerrhandle(login, err_handler) == NULL)
		ThrowError(CDBLIBERR::eDbProcHandler);

	if (dbprocmsghandle(login, msg_handler) == NULL)
		ThrowError(CDBLIBERR::eDbProcHandler);


	DBSETLUSER(login, szUser);
	DBSETLPWD(login, szPassword);
	DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
	DBSETLHOST(login, szHost);
	DBSETLVERSION(login, DBVER60);			//
use dblib ver 6.0 client behavior

	// set time to wait for login
	if (dbsetlogintime(60) == FAIL)
		ThrowError(CDBLIBERR::eDbSet);

	// set time to wait for statement execution
	if (dbsettime(180) == FAIL)
		ThrowError(CDBLIBERR::eDbSet);

	m_dbproc = dbopen(login, szServer);

	// deallocate login structure before checking for success
	dbfreelogin( login );

	if (m_dbproc == NULL)
		ThrowError(CDBLIBERR::eDbOpen);

	// save address of class instance so that the message and
error handler
	// can get to data.
	dbsetuserdata(m_dbproc, (LPVOID)this);

	// Use the the right database
	if (dbuse(m_dbproc, szDatabase) == FAIL)
		ThrowError(CDBLIBERR::eDbUse);

	dbcmd(m_dbproc, "set nocount on ");
	// do not return row counts
	dbcmd(m_dbproc, "set XACT_ABORT ON");
	// rollback transaction on abort

	if (dbsqlexec(m_dbproc) == FAIL)
		ThrowError(CDBLIBERR::eDbSqlExec);

	DiscardNextResults(2);

	// verify that version of stored procs on server is correct
	dbrpcinit(m_dbproc, "tpcc_version", 0);

	if (dbrpcexec(m_dbproc) == FAIL)
		ThrowError(CDBLIBERR::eDbRpcExec);

	if (dbresults(m_dbproc) != SUCCEED)
		ThrowError(CDBLIBERR::eDbResults);

	if (dbnextrow(m_dbproc) != REG_ROW)
		ThrowError(CDBLIBERR::eDbNextRow);

	char szSrvVersion[16];
	pData=dbdata(m_dbproc, 1);
```

```
		if (pData)
			UtilStrCpy(szSrvVersion, pData,
dbdatlen(m_dbproc, 1));
		else
			szSrvVersion[0]=0;
		if (strcmp(szSrvVersion,sVersion))
			throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );


		DiscardNextRows(0);
		DiscardNextResults(0);
}


CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
		// close db connection and deallocate resources
		dbclose(m_dbproc);
		InterlockedDecrement( &iConnectionCount );
		if (m_DbLibErr != NULL)
			delete m_DbLibErr;
		if (m_SqlErr != NULL)
			delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int dberr, int oserr,
LPCSTR dberrstr, LPCSTR oserrstr)
{
		delete m_DbLibErr;
		m_DbLibErr = new CDBLIBERR(CDBLIBERR::eUnknown,
severity, dberr, oserr);

		if (dberrstr != NULL)
		{
			m_DbLibErr->m_dberrstr = new char[
strlen(dberrstr)+1 ];
			strcpy( m_DbLibErr->m_dberrstr, dberrstr );
		}

		if (oserrstr != NULL)
		{
			m_DbLibErr->m_oserrstr = new char[
strlen(oserrstr)+1 ];
			strcpy( m_DbLibErr->m_oserrstr, oserrstr );
		}

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int
severity, LPCSTR msgtext )
{
		if (m_SqlErr == NULL)
			m_SqlErr = new CSQLERR();

		m_SqlErr->m_msgno = msgno;
		m_SqlErr->m_msgstate = msgstate;
		m_SqlErr->m_severity = severity;

		delete [] m_SqlErr->m_msgtext;
		if (msgtext != NULL)
		{
			m_SqlErr->m_msgtext = new char[
strlen(msgtext)+1 ];
			strcpy( m_SqlErr->m_msgtext, msgtext );
		}
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction )
{
		// discard anything still in return buffer
		DiscardNextRows(-1);
		DiscardNextResults(-1);

		// check for SQL Server error first;  if yes, throw it and
ignore any DBLib error.
		if (m_SqlErr != NULL)
		{
			CSQLERR		*pSqlErr;
			pSqlErr = m_SqlErr;
			m_SqlErr = NULL;		// clear our
pointer to instance; catch handler will delete
			throw pSqlErr;
		}

		CDBLIBERR  *pDbLibErr;
		if (m_DbLibErr == NULL)
			// this case isn't expected to happen, since it
means that an error was returned
			// but the error handlers were not called.
			pDbLibErr = new CDBLIBERR(eAction);
		else
		{
			pDbLibErr = m_DbLibErr;
			pDbLibErr->m_eAction = eAction;
			m_DbLibErr = NULL;			//
clear our pointer to instance; catch handler will delete
		}

		throw pDbLibErr;
}

// Read and discard rows until no more.  Throw an exception if number
of rows read doesn't
// match number of rows expected.  The row count will be ignored if
the expected count value
// passed in is negative.  A typical use of this routine is to verify that
there are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
		int			iRowsRead = 0;
		RETCODE	rc;

		while (TRUE)
		{
			rc = dbnextrow(m_dbproc);
			if (rc == NO_MORE_ROWS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)

		ThrowError(CDBLIBERR::eDbNextRow);
				else
					break;
			}
			iRowsRead++;
		}

		if ((iExpectedCount >= 0) &&
			(iExpectedCount != iRowsRead))
			ThrowError(CDBLIBERR::eWrongRowCount);
}

// Read and discard results until no more.  Throw an exception if
number of result sets read doesn't
// match number expected.  The result set count will be ignored if the
expected count value
// passed in is negative.  A typical use of this routine is to verify that
there are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
		int			iResultsRead = 0;
		RETCODE	rc;

		while (TRUE)
		{
			rc = dbresults(m_dbproc);
			if (rc == NO_MORE_RESULTS)
				break;
			if (rc == FAIL)
			{
				if (iExpectedCount >= 0)

		ThrowError(CDBLIBERR::eDbResults);
				else
					break;
			}
```

```
                        DiscardNextRows(-1);
                        iResultsRead++;
                }

                if ((iExpectedCount >= 0) &&
                        (iExpectedCount != iResultsRead))
                        ThrowError(CDBLIBERR::eWrongRowCount);
}

void CTPCC_DBLIB::StockLevel()
{
                int                                     iTryCount = 0;
                const BYTE   *pData;

                ResetError();

                while (TRUE)
                {
                        try
                        {
                                dbrpcinit(m_dbproc,
"tpcc_stocklevel", 0);

                                dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.StockLevel.w_id);
                // @w_id int
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.StockLevel.d_id);
                // @d_id tinyint
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.threshold);   //
@threshhold smallint

                                if (dbrpcexec(m_dbproc) ==
FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

                                if (dbresults(m_dbproc) !=
SUCCEED)

                ThrowError(CDBLIBERR::eDbResults);

                                if (dbnextrow(m_dbproc) !=
REG_ROW)

                ThrowError(CDBLIBERR::eDbNextRow);

                                if (pData=dbdata(m_dbproc,
1))

                m_txn.StockLevel.low_stock = *((long *) pData);
                                DiscardNextRows(0);
                                DiscardNextResults(0);


                m_txn.StockLevel.exec_status_code = eOK;
                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                        (e->m_msgno
== iErrOleDbProvider &&
                                        strstr(e-
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                        (++iTryCount <=
iMaxRetries))
                                {
                                        // hit deadlock;
backoff for increasingly longer period
                                        delete e;
                                        Sleep(10 *
iTryCount);
                                }
                                else
                                        throw;
                        }
                }               // while (TRUE)

                //if (iTryCount)
                //              throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}


void CTPCC_DBLIB::NewOrder()
{
                int                                     i;
                DBINT                   commit_flag;
                DBDATETIMEdatetime;
                DBDATEREC  daterec;

                int                                     iTryCount = 0;
                const BYTE   *pData;

                ResetError();

                while (TRUE)
                {
                        try
                        {
                                dbrpcinit(m_dbproc,
"tpcc_neworder", 0);
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.w_id);
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.d_id);
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.c_id);
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_ol_cnt);

                                // check whether any order
lines are for a remote warehouse
                                m_txn.NewOrder.o_all_local =
1;
                                for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
                                {
                                        if
(m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
                                        {

                m_txn.NewOrder.o_all_local = 0;  // at least one remote
warehouse

                break;
                                        }
                                }
                                dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

                                for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
                                {

                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_txn.NewOrder.OL[i].ol_i_id);

                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_txn.NewOrder.OL[i].ol_supply_w_id);

                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_txn.NewOrder.OL[i].ol_quantity);
                                }

                                if (dbrpcexec(m_dbproc) ==
FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);


                                // Get order line results
```

```
                                m_txn.NewOrder.total_amount
= 0;
                                for (i = 0;
i<m_txn.NewOrder.o_ol_cnt; i++)
                                {
                                        if
(dbresults(m_dbproc) != SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);
                                        if
(dbnumcols(m_dbproc) != 5)
        ThrowError(CDBLIBERR::eWrongNumCols);
                                        if
(dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);

        if(pData=dbdata(m_dbproc, 1))
        UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData,
dbdatlen(m_dbproc, 1));
        if(pData=dbdata(m_dbproc, 2))
        m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *)
pData);
        if(pData=dbdata(m_dbproc, 3))
        UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic,
pData, dbdatlen(m_dbproc, 3));
        if(pData=dbdata(m_dbproc, 4))
        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),
        SQLFLT8, (BYTE *)&m_txn.NewOrder.OL[i].ol_i_price, 8);
        if(pData=dbdata(m_dbproc, 5))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5),
        SQLFLT8, (BYTE *)&m_txn.NewOrder.OL[i].ol_amount,
8);
        m_txn.NewOrder.total_amount =
m_txn.NewOrder.total_amount + m_txn.NewOrder.OL[i].ol_amount;

        DiscardNextRows(0);
                                }

                                // get remaining values for
w_tax, d_tax, o_id, c_last, c_discount, c_credit, o_entry_d, commit_flag
                                if (dbresults(m_dbproc) !=
SUCCEED)
        ThrowError(CDBLIBERR::eDbResults);
                                if (dbnextrow(m_dbproc) !=
REG_ROW)
        ThrowError(CDBLIBERR::eDbNextRow);
                                if (dbnumcols(m_dbproc) != 8)
        ThrowError(CDBLIBERR::eWrongNumCols);
                                if (pData=dbdata(m_dbproc,
1))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,1), SQLFLT8, (BYTE *)&m_txn.NewOrder.w_tax, 8);
                                if (pData=dbdata(m_dbproc,
2))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,2), SQLFLT8, (BYTE *)&m_txn.NewOrder.d_tax, 8);
                                if (pData=dbdata(m_dbproc,
3))
        m_txn.NewOrder.o_id = (*(DBINT *) pData);
                                if (pData=dbdata(m_dbproc,
4))
        UtilStrCpy(m_txn.NewOrder.c_last, pData,
dbdatlen(m_dbproc, 4));
                                if (pData=dbdata(m_dbproc,
5))
        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);
                                if (pData=dbdata(m_dbproc,
6))
        UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));
                                if (pData=dbdata(m_dbproc,
7))
                                {
                                        datetime =
*((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.NewOrder.o_entry_d.year   = daterec.year;
        m_txn.NewOrder.o_entry_d.month  = daterec.month;
        m_txn.NewOrder.o_entry_d.day    = daterec.day;
        m_txn.NewOrder.o_entry_d.hour   = daterec.hour;
        m_txn.NewOrder.o_entry_d.minute = daterec.minute;
        m_txn.NewOrder.o_entry_d.second = daterec.second;
                                }
                                if (pData=dbdata(m_dbproc,
8))
                                        commit_flag =
(*(DBTINYINT *) pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);

                                if (commit_flag == 1)
                                {
        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));
        m_txn.NewOrder.exec_status_code = eOK;
                                }
                                else
        m_txn.NewOrder.exec_status_code = eInvalidItem;

                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                        (e->m_msgno
== iErrOleDbProvider &&
```

```
                                                strstr(e->m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                                (++iTryCount <= iMaxRetries))
                                        {
                                                // hit deadlock; backoff for increasingly longer period
                                                delete e;
                                                Sleep(10 * iTryCount);
                                        }
                                        else
                                                throw;
                        }
                }               // while (TRUE)

//              if (iTryCount)
//                      throw new CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS, iTryCount);
}


void CTPCC_DBLIB::Payment()
{
        DBDATETIMEdatetime;
        DBDATEREC  daterec;

        int                                     iTryCount = 0;
        const BYTE   *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc, "tpcc_payment", 0);

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *) &m_txn.Payment.h_amount);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.c_d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_id);

                        // if customer id is zero, then payment is by name
                        if (m_txn.Payment.c_id == 0)

                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1, strlen(m_txn.Payment.c_last), (unsigned char *)m_txn.Payment.c_last);

                        if (dbrpcexec(m_dbproc) == FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc) != SUCCEED)

                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc) != REG_ROW)

                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) != 27)

                ThrowError(CDBLIBERR::eWrongNumCols);

                        if (pData=dbdata(m_dbproc, 1))

                m_txn.Payment.c_id = *((DBINT *) pData);
                        if (pData=dbdata(m_dbproc, 2))

                UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
                        if (pData=dbdata(m_dbproc, 3))
                        {
                                datetime = *((DBDATETIME *) pData);

                dbdatecrack(m_dbproc, &daterec, &datetime);

                m_txn.Payment.h_date.year   = daterec.year;

                m_txn.Payment.h_date.month  = daterec.month;

                m_txn.Payment.h_date.day    = daterec.day;

                m_txn.Payment.h_date.hour   = daterec.hour;

                m_txn.Payment.h_date.minute = daterec.minute;
                m_txn.Payment.h_date.second = daterec.second;
                        if (pData=dbdata(m_dbproc, 4))

                UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc, 4));
                        if (pData=dbdata(m_dbproc, 5))

                UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc, 5));
                        if (pData=dbdata(m_dbproc, 6))

                UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
                        if (pData=dbdata(m_dbproc, 7))

                UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
                        if (pData=dbdata(m_dbproc, 8))

                UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
                        if (pData=dbdata(m_dbproc, 9))

                UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc, 9));
                        if (pData=dbdata(m_dbproc, 10))

                UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc, 10));
                        if (pData=dbdata(m_dbproc, 11))

                UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
                        if (pData=dbdata(m_dbproc, 12))

                UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc, 12));
                        if (pData=dbdata(m_dbproc, 13))

                UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
```

```
                                if (pData=dbdata(m_dbproc,
14))

        UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));
                                if (pData=dbdata(m_dbproc,
15))

        UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
                                if (pData=dbdata(m_dbproc,
16))

        UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
                                if (pData=dbdata(m_dbproc,
17))

        UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));
                                if (pData=dbdata(m_dbproc,
18))

        UtilStrCpy(m_txn.Payment.c_city, pData,
dbdatlen(m_dbproc, 18));
                                if (pData=dbdata(m_dbproc,
19))

        UtilStrCpy(m_txn.Payment.c_state, pData,
dbdatlen(m_dbproc, 19));
                                if (pData=dbdata(m_dbproc,
20))

        UtilStrCpy(m_txn.Payment.c_zip, pData,
dbdatlen(m_dbproc, 20));
                                if (pData=dbdata(m_dbproc,
21))

        UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));
                                if (pData=dbdata(m_dbproc,
22))
                                {
                                        datetime =
*((DBDATETIME *) pData);

        dbdatecrack(m_dbproc, &daterec, &datetime);

        m_txn.Payment.c_since.year   = daterec.year;

        m_txn.Payment.c_since.month  = daterec.month;

        m_txn.Payment.c_since.day    = daterec.day;
        m_txn.Payment.c_since.hour   = daterec.hour;

        m_txn.Payment.c_since.minute = daterec.minute;

        m_txn.Payment.c_since.second = daterec.second;
                                }
                                if(pData=dbdata(m_dbproc,
23))

        UtilStrCpy(m_txn.Payment.c_credit, pData,
dbdatlen(m_dbproc, 23));
                                if(pData=dbdata(m_dbproc,
24))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
                                if(pData=dbdata(m_dbproc,
25))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
                                if(pData=dbdata(m_dbproc,
26))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
                                if(pData=dbdata(m_dbproc,
27))

        UtilStrCpy(m_txn.Payment.c_data, pData,
dbdatlen(m_dbproc, 27));

                                DiscardNextRows(0);
                                DiscardNextResults(0);


                                if (m_txn.Payment.c_id == 0)
                                        throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                                else

        m_txn.Payment.exec_status_code = eOK;

                                return;
                        }
                        catch (CSQLERR *e)
                        {
                                if ((e->m_msgno == 1205 ||
                                                        (e->m_msgno
== iErrOleDbProvider &&
                                                        strstr(e-
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                                        (++iTryCount <=
iMaxRetries))
                                        {
                                                        // hit deadlock;
backoff for increasingly longer period
                                                        delete e;
                                                        Sleep(10 *
iTryCount);
                                        }
                                        else
                                                        throw;
                        }
                }               // while (TRUE)

//              if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}



void CTPCC_DBLIB::OrderStatus()
{
        int                             i;
        DBDATETIMEdatetime;
        DBDATEREC  daterec;

        int                             iTryCount = 0;
        RETCODE                 rc;
        const BYTE   *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc,
"tpcc_orderstatus", 0);

                        dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.OrderStatus.w_id);
                        dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.OrderStatus.d_id);
                        dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.OrderStatus.c_id);

                        // if customer id is zero, then
order status is by name
```

```
                                if (m_txn.OrderStatus.c_id ==
0)

        dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
strlen(m_txn.OrderStatus.c_last), (unsigned char
*)m_txn.OrderStatus.c_last);


                                if (dbrpcexec(m_dbproc) ==
FAIL)

        ThrowError(CDBLIBERR::eDbRpcExec);

                                // Get order lines
                                if (dbresults(m_dbproc) !=
SUCCEED)
                                {
                                        if ((m_DbLibErr
== NULL) && (m_SqlErr == NULL))

        throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                        else

        ThrowError(CDBLIBERR::eDbResults);
                                }

                                if (dbnumcols(m_dbproc) != 5)

        ThrowError(CDBLIBERR::eWrongNumCols);

                                i = 0;
                                while (TRUE)
                                {
                                        rc =
dbnextrow(m_dbproc);
                                        if (rc ==
NO_MORE_ROWS)

        break;
                                        if (rc !=
REG_ROW)

        ThrowError(CDBLIBERR::eDbNextRow);


        if(pData=dbdata(m_dbproc, 1))

        m_txn.OrderStatus.OL[i].ol_supply_w_id =
(*(DBSMALLINT *) pData);

        if(pData=dbdata(m_dbproc, 2))

        m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
        if(pData=dbdata(m_dbproc, 3))

        m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *)
pData);

        if(pData=dbdata(m_dbproc, 4))

        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),

                        SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);

        if(pData=dbdata(m_dbproc, 5))
                                        {

        datetime = *((DBDATETIME *) pData);

        dbdatecrack(m_dbproc, &daterec, &datetime);

        m_txn.OrderStatus.OL[i].ol_delivery_d.year   =
daterec.year;

        m_txn.OrderStatus.OL[i].ol_delivery_d.month  =
daterec.month;

        m_txn.OrderStatus.OL[i].ol_delivery_d.day    =
daterec.day;

        m_txn.OrderStatus.OL[i].ol_delivery_d.hour   =
daterec.hour;

        m_txn.OrderStatus.OL[i].ol_delivery_d.minute =
daterec.minute;

        m_txn.OrderStatus.OL[i].ol_delivery_d.second =
daterec.second;
                                        }
                                        i++;
                                }
                                m_txn.OrderStatus.o_ol_cnt =
i;


                                if (dbresults(m_dbproc) !=
SUCCEED)

        ThrowError(CDBLIBERR::eDbResults);

                                if (dbnextrow(m_dbproc) !=
REG_ROW)

        ThrowError(CDBLIBERR::eDbNextRow);
                                if (dbnumcols(m_dbproc) != 8)

        ThrowError(CDBLIBERR::eWrongNumCols);

                                if(pData=dbdata(m_dbproc,
1))

        m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc,
2))

        UtilStrCpy(m_txn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
                                if(pData=dbdata(m_dbproc,
3))

        UtilStrCpy(m_txn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
                                if(pData=dbdata(m_dbproc,
4))

        UtilStrCpy(m_txn.OrderStatus.c_middle, pData,
dbdatlen(m_dbproc, 4));
                                if(pData=dbdata(m_dbproc,
5))
                                {
                                        datetime =
*((DBDATETIME *) pData);

        dbdatecrack(m_dbproc, &daterec, &datetime);

        m_txn.OrderStatus.o_entry_d.year   = daterec.year;

        m_txn.OrderStatus.o_entry_d.month  = daterec.month;

        m_txn.OrderStatus.o_entry_d.day    = daterec.day;

        m_txn.OrderStatus.o_entry_d.hour   = daterec.hour;

        m_txn.OrderStatus.o_entry_d.minute = daterec.minute;

        m_txn.OrderStatus.o_entry_d.second = daterec.second;
                                }
                                if(pData=dbdata(m_dbproc,
6))

        m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *)
pData);
                                if(pData=dbdata(m_dbproc,
7))
```

```
        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,7),
          SQLFLT8, (BYTE *)&m_txn.OrderStatus.c_balance, 8);
                        if(pData=dbdata(m_dbproc,
8))
        m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


                        if (m_txn.OrderStatus.o_ol_cnt
== 0)
                                throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                        else if (m_txn.OrderStatus.c_id
== 0 && m_txn.OrderStatus.c_last[0] == 0)
                                throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                        else

        m_txn.OrderStatus.exec_status_code = eOK;

                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205 ||
                                (e->m_msgno
== iErrOleDbProvider &&
                                strstr(e-
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                (++iTryCount <=
iMaxRetries))
                        {
                                // hit deadlock;
backoff for increasingly longer period
                                delete e;
                                Sleep(10 *
iTryCount);
                        }
                        else
                                throw;
                }
        }       // while (TRUE)

//      if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);


void CTPCC_DBLIB::Delivery()
{
        int                             i;
        int                             iTryCount = 0;
        const BYTE   *pData;

        ResetError();

        while (TRUE)
        {
                try
                {
                        dbrpcinit(m_dbproc,
"tpcc_delivery", 0);

                        dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_txn.Delivery.w_id);
                        dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

                        if (dbrpcexec(m_dbproc) ==
FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

                        if (dbresults(m_dbproc) !=
SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

                        if (dbnextrow(m_dbproc) !=
REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);

                        if (dbnumcols(m_dbproc) !=
10)
                ThrowError(CDBLIBERR::eWrongNumCols);

                        for (i=0; i<10; i++)
                        {
                                if (pData =
dbdata(m_dbproc, i+1))
                m_txn.Delivery.o_id[i] = *((DBINT *)pData);
                        }
                        DiscardNextRows(0);
                        DiscardNextResults(0);


                m_txn.Delivery.exec_status_code = eOK;
                        return;
                }
                catch (CSQLERR *e)
                {
                        if ((e->m_msgno == 1205 ||
                                (e->m_msgno
== iErrOleDbProvider &&
                                strstr(e-
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                                (++iTryCount <=
iMaxRetries))
                        {
                                // hit deadlock;
backoff for increasingly longer period
                                delete e;
                                Sleep(10 *
iTryCount);
                        }
                        else
                                throw;
                }
        }       // while (TRUE)

//      if (iTryCount)
//                      throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::ResetError()
{
        if (m_DbLibErr != NULL)
        {
                delete m_DbLibErr;
                m_DbLibErr = (CDBLIBERR*)NULL;
        }

        if (m_SqlErr != NULL)
        {
                delete m_SqlErr;
                m_SqlErr = (CSQLERR*)NULL;
        }
        return;
}
```

Db dblib tpcc_dblib.h

```
/*              FILE:                   TPCC_DBLIB.H
 *                                              Microsoft TPC-C
Kit Ver. 4.20.000
 *                                              Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                              Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:        Header file for TPC-C txn class
implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void   // dbprocess structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
        public:

                CSQLERR(void)
                {
                        m_msgno = 0;
                        m_msgstate = 0;
                        m_severity = 0;
                        m_msgtext = NULL;
                };

                ~CSQLERR()
                {
                        delete [] m_msgtext;
                };

                int             m_msgno;
                int             m_msgstate;
                int             m_severity;
                char   *m_msgtext;
                int             ErrorType()
{return ERR_TYPE_SQL;};
                char*   ErrorTypeStr() { return "SQL";
}
                int             ErrorNum()
{return m_msgno;};
                char*   ErrorText() {return
m_msgtext;};

};

class CDBLIBERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eLogin,
                // error from dblogin
                        eDbOpen,
        // error from dbopen
                        eDbUse,
                // error from dbuse
                        eDbSqlExec,
        // error from dbsqlexec
                        eDbSet,
                // error from one of the dbset* routines
                        eDbNextRow,
        // error from dbnextrow
                        eWrongRowCount,
        // more or less rows returned than expected
                        eWrongNumCols,
        // more or less columns returned than expected
                        eDbResults,
        // error from dbresults
                        eDbRpcExec,
        // error from dbrpcexec
                        eDbSetMaxProcs,
        // error from dbsetmaxprocs
                        eDbProcHandler
        // error from either dbprocerrhandle or dbprocmsghandle
                };

                CDBLIBERR(ACTION eAction, int severity =
0, int dberror = 0, int oserr = 0)
                {
                        m_eAction = eAction;
                        m_severity = severity;
                        m_dberror = dberror;
                        m_oserr = oserr;
                        m_dberrstr = NULL;
                        m_oserrstr = NULL;
                };

                ~CDBLIBERR()
                {
                        delete [] m_dberrstr;
                        delete [] m_oserrstr;
                };

                ACTION  m_eAction;
                int             m_severity;
                int             m_dberror;
                int             m_oserr;
                char   *m_dberrstr;
                char   *m_oserrstr;

                int             ErrorType()
{return ERR_TYPE_DBLIB;};
                char*   ErrorTypeStr() { return
"DBLIB"; }
                int             ErrorNum()
{return m_dberror;};
                char*   ErrorText() {return
m_dberrstr;};
                int             ErrorAction() {
return (int)m_eAction; }

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
        public:
                enum CTPCC_DBLIB_ERRS
                {
                        ERR_WRONG_SP_VERSION =
1,      // "Wrong version of stored procs on database server"
                        ERR_INVALID_CUST,
                // "Invalid Customer id,name."
                        ERR_NO_SUCH_ORDER,
                // "No orders found for customer."
                        ERR_RETRIED_TRANS,
                // "Retries before transaction succeeded."
                };

                CTPCC_DBLIB_ERR( int iErr ) { m_errno =
iErr; m_iTryCount = 0; };

                CTPCC_DBLIB_ERR( int iErr, int iTryCount ) {
m_errno = iErr; m_iTryCount = iTryCount; };

                int             m_errno;
                int             m_iTryCount;
```

```
                        int                     ErrorType()
{return ERR_TYPE_TPCC_DBLIB;};
                        char*           ErrorTypeStr() { return "TPCC
DBLIB"; }
                        int                     ErrorNum()
{return m_errno;};

                        char*           ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
        private:
                // declare variables and private functions
here...
                PDBPROCESSm_dbproc;
                CDBLIBERR   *m_DbLibErr;
        // not allocated until needed (maybe never)
                CSQLERR                 *m_SqlErr;
                // not allocated until needed (maybe never)
                int
        m_MaxRetries;                   // retry count on
deadlock

                void DiscardNextRows(int iExpectedCount);
                void DiscardNextResults(int iExpectedCount);
                void ThrowError( CDBLIBERR::ACTION
eAction );
                void ResetError();

                union
                {
                        NEW_ORDER_DATA
        NewOrder;
                        PAYMENT_DATA
        Payment;
                        DELIVERY_DATA
        Delivery;
                        STOCK_LEVEL_DATA
        StockLevel;
                        ORDER_STATUS_DATA
        OrderStatus;
                }
        m_txn;

        public:
                CTPCC_DBLIB(LPCSTR szServer, LPCSTR
szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );
                ~CTPCC_DBLIB(void);
```

```
                        inline PNEW_ORDER_DATA
        BuffAddr_NewOrder()             { return
&m_txn.NewOrder;    };  inline PPAYMENT_DATA
        BuffAddr_Payment()              { return
&m_txn.Payment;     };
                        inline PDELIVERY_DATA
        BuffAddr_Delivery()             { return
&m_txn.Delivery;    };
                        inline PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()   { return &m_txn.StockLevel;
};
                        inline PORDER_STATUS_DATA
        BuffAddr_OrderStatus()  { return &m_txn.OrderStatus;
};

                        void NewOrder                   ();
                        void Payment            ();
                        void Delivery           ();
                        void StockLevel                 ();
                        void OrderStatus        ();


                // these are public because they must be
called from the dblib err_handler and msg_hangler
                // outside of the class
                void SetDbLibError(int severity, int dberr, int
oserr, LPCSTR dberrstr, LPCSTR oserrstr);
                void SetSqlError( int msgno, int msgstate, int
severity, LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase );

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR);
```

db_odbc.dll tpcc_odbc.dll

```
/*      FILE:           TPCC_ODBC.CPP
 *                              Microsoft TPC-C
Kit Ver. 4.42.000
 *                              Copyright
Microsoft, 2002
 *      All Rights Reserved
 *                                              Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *      PURPOSE:    Implements ODBC calls for TPC-C txns.
 *      Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.42.000 - changed w_id fields from short to
long to support >32K warehouses
 *                  4.20.000 - updated rev number to match kit
 *                  4.10.001 - not deleting error class in catch
handler on deadlock retry;
 *                                                  not a functional
bug, but a memory leak
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

//#define COMPILE_FOR_SNAC      // define that to compile for
SQL Native Client; comment out to use MDAC

#ifndef COMPILE_FOR_SNAC
#include <odbcss.h>
#else
// Compile for SNAC
#include <sqlncli.h>
#endif

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_odbc.h"

// version string; must match return value from tpcc_version stored
proc
const char    sVersion[] = "4.20.000";

const iMaxRetries = 10;                 // how many retries on
deadlock
```

```
const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

static SQLHENV henv = SQL_NULL_HENV;
                        // ODBC environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
        switch( ul_reason_for_call )
        {
                case DLL_PROCESS_ATTACH:

        DisableThreadLibraryCalls(hModule);
                        if (
SQLAllocHandleStd(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv) !=
SQL_SUCCESS )
                                return FALSE;
                        break;

                case DLL_PROCESS_DETACH:
                        if (henv != NULL)

        SQLFreeEnv(henv);
                        break;

                default:
                        /* nothing */;
        }
        return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

char* CTPCC_ODBC_ERR::ErrorText(void)
{
        int i;

        static SERRORMSG errorMsgs[] =
        {
                { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database server"   },
                { ERR_INVALID_CUST,
        "Invalid Customer id,name."
                                                        },
                { ERR_NO_SUCH_ORDER,                "No
orders found for customer."
                        },
                { ERR_RETRIED_TRANS,
        "Retries before transaction succeeded."
                { 0,
                                ""


        }
        };

        static char szNotFound[] = "Unknown error number.";

        for(i=0; errorMsgs[i].szMsg[0]; i++)
        {
                if ( m_errno == errorMsgs[i].iError )
                        break;
        }
        if ( !errorMsgs[i].szMsg[0] )
                return szNotFound;
        else
                return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
        LPCSTR szServer,                // name of SQL
server
        LPCSTR szUser,                                  //
user name for login
        LPCSTR szPassword,              // password for
login
        LPCSTR szHost,                                  //
not used
        LPCSTR szDatabase,              // name of
database to use
        LPCWSTR szSPPrefix,     // prefix to append to the
stored procedure names
        BOOL bCallNoDuplicatesNewOrder ) // whether to check
for non-duplicate items in NewOrder and call a new SP
{
        return new CTPCC_ODBC( szServer, szUser, szPassword,
szHost, szDatabase, szSPPrefix, bCallNoDuplicatesNewOrder );
}


CTPCC_ODBC::CTPCC_ODBC (
        LPCSTR szServer,
        // name of SQL server
        LPCSTR szUser,
                // user name for login
        LPCSTR szPassword,
        // password for login
        LPCSTR szHost,
                // not used
        LPCSTR szDatabase,
        // name of database to use
        LPCWSTR szSPPrefix,
        // prefix to append to the stored procedure names
        BOOL    bCallNoDuplicatesNewOrder // whether to
check for non-duplicate items in NewOrder and call a new SP
        )
: m_bCallNoDuplicatesNewOrder(bCallNoDuplicatesNewOrder)
{
        RETCODE                 rc;

        // initialization
        m_hdbc = SQL_NULL_HDBC;
        m_hstmt = SQL_NULL_HSTMT;

        m_hstmtNewOrder = SQL_NULL_HSTMT;
        m_hstmtPayment = SQL_NULL_HSTMT;
        m_hstmtDelivery = SQL_NULL_HSTMT;
        m_hstmtOrderStatus = SQL_NULL_HSTMT;
        m_hstmtStockLevel = SQL_NULL_HSTMT;

        m_descNewOrderCols1 = SQL_NULL_HDESC;
        m_descNewOrderCols2 = SQL_NULL_HDESC;
        m_descOrderStatusCols1 = SQL_NULL_HDESC;
        m_descOrderStatusCols2 = SQL_NULL_HDESC;

        wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));


        if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc)
!= SQL_SUCCESS )
                ThrowError(CODBCERR::eAllocHandle);

        if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE,
4096) != SQL_SUCCESS )
                ThrowError(CODBCERR::eConnOption);

        {
                char
        szConnectStr[256];
                char
        szOutStr[1024];
                SQLSMALLINT
        iOutStrLen;

#ifndef COMPILE_FOR_SNAC
                sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                        szServer, szUser, szPassword,
szDatabase );
#else
                // Compile for SNAC
```

```
                        sprintf( szConnectStr, "DRIVER=SQL Native
Client;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                                szServer, szUser, szPassword,
szDatabase );
#endif
                        rc = SQLDriverConnect(m_hdbc, NULL,
(SQLCHAR*)szConnectStr, sizeof(szConnectStr),
                                (SQLCHAR*)szOutStr,
sizeof(szOutStr), &iOutStrLen, SQL_DRIVER_NOPROMPT );

                        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eConnect);
                }

                if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmt) != SQL_SUCCESS)
                        ThrowError(CODBCERR::eAllocHandle);

                {
                        char                    buffer[128];

                        // set some options affecting connection
behavior
                        strcpy(buffer, "set nocount on set
XACT_ABORT ON");
                        rc = SQLExecDirect(m_hstmt, (unsigned
char *)buffer, SQL_NTS);
                        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

                        // verify that version of stored procs on
server is correct
                        char db_sp_version[10];
                        strcpy(buffer, "{call tpcc_version}");
                        rc = SQLExecDirect(m_hstmt, (unsigned
char *)buffer, SQL_NTS);
                        if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);
                        if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR,
&db_sp_version, sizeof(db_sp_version), NULL) != SQL_SUCCESS )

                ThrowError(CODBCERR::eBindCol);
                        if ( SQLFetch(m_hstmt) == SQL_ERROR )

                ThrowError(CODBCERR::eFetch);
                        if (strcmp(db_sp_version,sVersion))
                                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );
                                SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmt);
                }

                // Bind parameters for each of the transactions
                InitNewOrderParams();
                InitPaymentParams();
                InitOrderStatusParams();
                InitDeliveryParams();
                InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
                // note: descriptors are automatically released when the
connection is dropped
                SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
                SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
                SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
                SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtOrderStatus);
                SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtStockLevel);

                SQLDisconnect(m_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC, m_hdbc);
}

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
                RETCODE                 rc;
                SDWORD                  lNativeError;
                char                    szState[6];
                char
                szMsg[SQL_MAX_MESSAGE_LENGTH];
                char
                szTmp[6*SQL_MAX_MESSAGE_LENGTH];
                CODBCERR   *pODBCErr;                           //
not allocated until needed (maybe never)

                pODBCErr = new CODBCERR();

                pODBCErr->m_NativeError = 0;
                pODBCErr->m_eAction = eAction;
                pODBCErr->m_bDeadLock = FALSE;

                szTmp[0] = 0;
                while (TRUE)
                {
                        rc = SQLError(henv, m_hdbc, m_hstmt,
(BYTE *)&szState, &lNativeError,
(BYTE *)&szMsg, sizeof(szMsg), NULL);
                        if (rc == SQL_NO_DATA)
                                break;

                        // check for deadlock
                        if (lNativeError == 1205 || (lNativeError ==
iErrOleDbProvider &&
                                strstr(szMsg,
sErrTimeoutExpired) != NULL))
                                pODBCErr->m_bDeadLock =
TRUE;

                        // capture the (first) database error
                        if (pODBCErr->m_NativeError == 0 &&
lNativeError != 0)
                                pODBCErr->m_NativeError =
lNativeError;

                        // quit if there isn't enough room to
concatenate error text
                        if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
                                break;

                        // include line break after first error msg
                        if (szTmp[0] != 0)
                                strcat( szTmp, "\n");
                        strcat( szTmp, szMsg );
                }

                if (pODBCErr->m_odbcerrstr != NULL)
                {
                        delete [] pODBCErr->m_odbcerrstr;
                        pODBCErr->m_odbcerrstr = NULL;
                }

                if (strlen(szTmp) > 0)
                {
                        pODBCErr->m_odbcerrstr = new char[
strlen(szTmp)+1 ];
                        strcpy( pODBCErr->m_odbcerrstr, szTmp );
                }

                SQLFreeStmt(m_hstmt, SQL_CLOSE);
                throw pODBCErr;
}

void CTPCC_ODBC::InitStockLevelParams()
{
                if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtStockLevel) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eAllocHandle);
```

```
		m_hstmt = m_hstmtStockLevel;

		int i = 0;
		if (   SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.StockLevel.w_id, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
			)
			ThrowError(CODBCERR::eBindParam);

		if ( SQLBindCol(m_hstmt, 1, SQL_C_SLONG,
&m_txn.StockLevel.low_stock, 0, NULL) != SQL_SUCCESS )
			ThrowError(CODBCERR::eBindCol);

		//Compose Stock Level statement
		_snwprintf(m_szStockLevelCommand,
sizeof(m_szStockLevelCommand)/sizeof(m_szStockLevelCommand[0]),
			L"{call %stpcc_stocklevel (?,?,?)}",
m_szSPPrefix);
}

void CTPCC_ODBC::StockLevel()
{
		RETCODE			rc;
		int					iTryCount = 0;

		m_hstmt = m_hstmtStockLevel;

		while (TRUE)
		{
			try
			{
				rc =
SQLExecDirectW(m_hstmt, m_szStockLevelCommand, SQL_NTS);
				if (rc != SQL_SUCCESS && rc
!= SQL_SUCCESS_WITH_INFO)

		ThrowError(CODBCERR::eExecDirect);

				if ( SQLFetch(m_hstmt) ==
SQL_ERROR )

		ThrowError(CODBCERR::eFetch);

				SQLFreeStmt(m_hstmt,
SQL_CLOSE);

		m_txn.StockLevel.exec_status_code = eOK;
				break;
			}
			catch (CODBCERR *e)
			{
				if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
					throw;

				// hit deadlock; backoff for
increasingly longer period
				delete e;
				Sleep(10 * iTryCount);
			}
		}

//		if (iTryCount)
//			throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitNewOrderParams()
{
		if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrder) != SQL_SUCCESS
			|| SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtNewOrderNoDuplicates) != SQL_SUCCESS
			|| SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descNewOrderCols1) != SQL_SUCCESS
			|| SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descNewOrderCols2) != SQL_SUCCESS
			|| SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descNewOrderNoDuplicatesCols1) != SQL_SUCCESS
			|| SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descNewOrderNoDuplicatesCols2) != SQL_SUCCESS
			)
			ThrowError(CODBCERR::eAllocHandle);

		m_hstmt = m_hstmtNewOrder;

		if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols1, SQL_IS_POINTER
) != SQL_SUCCESS )
			ThrowError(CODBCERR::eSetStmtAttr);

		int i = 0;
		if (   SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.w_id, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.d_id, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.c_id, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_ol_cnt, 0, NULL) != SQL_SUCCESS
			|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
			)
			ThrowError(CODBCERR::eBindParam);

		for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
		{
			if (   SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) != SQL_SUCCESS
				|| SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
				|| SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.NewOrder.OL[j].ol_quantity, 0, NULL) != SQL_SUCCESS
				)

		ThrowError(CODBCERR::eBindParam);
		}

		// set the bind offset pointer
		if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_ROW_BIND_OFFSET_PTR, &m_BindOffset,
SQL_IS_POINTER ) != SQL_SUCCESS )
			ThrowError(CODBCERR::eSetStmtAttr);

		i = 0;
		if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) != SQL_SUCCESS
			|| SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0, NULL) !=
SQL_SUCCESS
			|| SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) !=
SQL_SUCCESS
			|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0, NULL) !=
SQL_SUCCESS
			|| SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0, NULL) !=
SQL_SUCCESS
			)
			ThrowError(CODBCERR::eBindCol);
```

```
        // associate the column bindings for the second result set
        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2, SQL_IS_POINTER
) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.w_tax, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE,    &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG,     &m_txn.NewOrder.o_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_last, sizeof(m_txn.NewOrder.c_last), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE,    &m_txn.NewOrder.c_discount, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.c_credit, sizeof(m_txn.NewOrder.c_credit), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG,     &m_no_commit_flag, 0, NULL) != SQL_SUCCESS
              )
                ThrowError(CODBCERR::eBindCol);

        //Compose the New Order statement
        _snwprintf(m_szNewOrderCommand,
sizeof(m_szNewOrderCommand)/sizeof(m_szNewOrderCommand[0]),
                  // 0         1         2
                  // 012345678901234567890123456789
                        L"{call
%stpcc_neworder(?,?,?,
```

```
		_snwprintf(m_szNewOrderNoDuplicatesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNewOrderNo
DuplicatesCommand[0]),
					L"{call
%stpcc_neworder_new(?,?,?,
```

```
                                if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                                if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

        ThrowError(CODBCERR::eFetch);

                                SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                                if (m_no_commit_flag == 1)
                                {

        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

        m_txn.NewOrder.exec_status_code = eOK;
                                }
                                else

        m_txn.NewOrder.exec_status_code = eInvalidItem;

                                break;
                        }
                        catch (CODBCERR *e)
                        {
                                if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                        throw;

                                // hit deadlock; backoff for
increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
        }

//      if (iTryCount)
//              throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

//
//      No lineitem duplicates optimized version.
//
void CTPCC_ODBC::NewOrderNoDuplicates()
{       int
        i;
        RETCODE                                         rc;
        int
        iTryCount = 0;

                                                        // 0
1       2       3

                                                        //
01234567890123456789012345678901234567890123
        wchar_t
        szSqlTemplate[iMAX_SP_NAME_LEN];

                                                        //= L"{call
tpcc_neworder_new(?,?,?,?,?,?,"

                                                        //
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"

                                                        //
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,"

                                                        //
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}";

        m_hstmt = m_hstmtNewOrderNoDuplicates;

        // associate the parameter and column bindings for this
transaction
        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

        // clip statement buffer based on number of parameters
        // fixed part is 33 chars and variable part is 6 chars per
line item
        wcscpy(szSqlTemplate,
m_szNewOrderNoDuplicatesCommand);
        i = m_iBeginNewOrderNoDuplicatesVariablePart +
m_txn.NewOrder.o_ol_cnt*6;
        wcscpy( &szSqlTemplate[i], L")}" );

        // check whether any order lines are for a remote
warehouse
        m_txn.NewOrder.o_all_local = 1;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
                if (m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
                {
                        m_txn.NewOrder.o_all_local =
0;  // at least one remote warehouse
                        break;
                }
        }

        while (TRUE)
        {
                try
                {
                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1, 0) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                        rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
                        if (rc != SQL_SUCCESS && rc
!= SQL_SUCCESS_WITH_INFO)

        ThrowError(CODBCERR::eExecDirect);

                        // configure block cursor
                        if (SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_NEW_ORDER_ITEMS, 0) != SQL_SUCCESS)

        ThrowError(CODBCERR::eSetStmtAttr);

                        // Get order line results

                        if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

        ThrowError(CODBCERR::eFetch);


                        m_txn.NewOrder.total_amount
= 0;
                        for (i = 0;
i<m_txn.NewOrder.o_ol_cnt; i++)
                        {


        m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
                        }

                        // associate the column
bindings for the second result set
```

```
                                if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                                // move to the next resultset
                                if ( SQLMoreResults(m_hstmt)
== SQL_ERROR )

        ThrowError(CODBCERR::eMoreResults);

                                if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

        ThrowError(CODBCERR::eFetch);

                                SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                                if (m_no_commit_flag == 1)
                                {

        m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

        m_txn.NewOrder.exec_status_code = eOK;
                                }
                                else

        m_txn.NewOrder.exec_status_code = eInvalidItem;

                                break;
                        }
                        catch (CODBCERR *e)
                        {
                                if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                        throw;

                                // hit deadlock; backoff for
increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
        }

//      if (iTryCount)
//                              throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);

void CTPCC_ODBC::InitPaymentParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtPayment) != SQL_SUCCESS )
                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtPayment;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
&m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.Payment.c_id,                    0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_last,                  sizeof(m_txn.Payment.c_last),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.h_date,            0,
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_1,
        sizeof(m_txn.Payment.w_street_1), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2,
        sizeof(m_txn.Payment.w_street_2), NULL) !=
SQL_SUCCESS      || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city,          sizeof(m_txn.Payment.w_city),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state,
        sizeof(m_txn.Payment.w_state), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,           sizeof(m_txn.Payment.w_zip),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1,
        sizeof(m_txn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2,
        sizeof(m_txn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,          sizeof(m_txn.Payment.d_city),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,
        sizeof(m_txn.Payment.d_state), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,           sizeof(m_txn.Payment.d_zip),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,         sizeof(m_txn.Payment.c_first),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,
        sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1,
        sizeof(m_txn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2,
        sizeof(m_txn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,          sizeof(m_txn.Payment.c_city),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,         sizeof(m_txn.Payment.c_state),
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,           sizeof(m_txn.Payment.c_zip),
NULL) != SQL_SUCCESS
```

```
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_phone,
        sizeof(m_txn.Payment.c_phone), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.c_since,           0,
NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_credit, sizeof(m_txn.Payment.c_credit), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE,   &m_txn.Payment.c_credit_lim,0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE,   &m_txn.Payment.c_discount,      0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE,   &m_txn.Payment.c_balance,       0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_data,          sizeof(m_txn.Payment.c_data),
NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);

        //Compose Payment statement
        _snwprintf(m_szPaymentCommand,
sizeof(m_szPaymentCommand)/sizeof(m_szPaymentCommand[0]),
                        L"{call %stpcc_payment (?,?,?,?,?,?,?)}",
m_szSPPrefix);
}

void CTPCC_ODBC::Payment()
{
        RETCODE                 rc;
        int                             iTryCount = 0;

        m_hstmt = m_hstmtPayment;

        if (m_txn.Payment.c_id != 0)
                m_txn.Payment.c_last[0] = 0;

        while (TRUE)
        {
                try
                {
                        rc =
SQLExecDirectW(m_hstmt, m_szPaymentCommand, SQL_NTS);
                        if (rc != SQL_SUCCESS && rc
!= SQL_SUCCESS_WITH_INFO)

        ThrowError(CODBCERR::eExecDirect);
```

```
                        if ( SQLFetch(m_hstmt) ==
SQL_ERROR)
        ThrowError(CODBCERR::eFetch);

                        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                        if (m_txn.Payment.c_id == 0)
                                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                        else

        m_txn.Payment.exec_status_code = eOK;

                        break;
                }
                catch (CODBCERR *e)
                {
                        if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for
increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }

//      if (iTryCount)
//              throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitOrderStatusParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtOrderStatus) != SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descOrderStatusCols1) != SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC,
m_hdbc, &m_descOrderStatusCols2) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtOrderStatus;

        if (   SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);
```

```
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.OrderStatus.c_last), 0, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

        // configure block cursor
        if (   SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
                || SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS
                )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.OL[0].ol_delivery_d, 0,
NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);


        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
```

```
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle),
NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.o_entry_d, 0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.o_carrier_id, 0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.c_balance, 0, NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);

        //Compose Order Status statement
        _snwprintf(m_szOrderStatusCommand,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOrderStatusCommand[0]
),
                        L"{call %stpcc_orderstatus (?,?,?,?)}",
m_szSPPrefix);
}

void CTPCC_ODBC::OrderStatus()
{
        int
                iTryCount = 0;
        RETCODE
        rc;

        m_hstmt = m_hstmtOrderStatus;

        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

        if (m_txn.OrderStatus.c_id != 0)
                m_txn.OrderStatus.c_last[0] = 0;

        while (TRUE)
        {
                try
                {
                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1, 0) != SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAttr);

                        rc =
SQLExecDirectW(m_hstmt, m_szOrderStatusCommand, SQL_NTS);
                        if ( ((rc ==
SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) || (rc ==
SQL_ERROR) )

                ThrowError(CODBCERR::eExecDirect);

                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLFetchScroll( m_hstmt,
SQL_FETCH_NEXT, 0 );
                        if ( ((rc ==
SQL_SUCCESS_WITH_INFO) && (m_RowsFetched != 0)) || (rc ==
SQL_ERROR) )

                ThrowError(CODBCERR::eFetchScroll);

                        m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

                        if (m_txn.OrderStatus.o_ol_cnt
!= 0)
                        {
                                if (
SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAttr);

                                if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

                ThrowError(CODBCERR::eMoreResults);

                                if ( (rc =
SQLFetch(m_hstmt)) == SQL_ERROR)

                ThrowError(CODBCERR::eFetch);
                        }

                        SQLFreeStmt(m_hstmt,
SQL_CLOSE);
                        if (m_txn.OrderStatus.o_ol_cnt
== 0)
                                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_NO_SUCH_ORDER );
                        else if (m_txn.OrderStatus.c_id
== 0 && m_txn.OrderStatus.c_last[0] == 0)
                                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                        else

                m_txn.OrderStatus.exec_status_code = eOK;

                        break;
                }
                catch (CODBCERR *e)
                {
                        if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for
increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }

//      if (iTryCount)
//              throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitDeliveryParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtDelivery) != SQL_SUCCESS )
                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtDelivery;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

        for (i=0;i<10;i++)
        {
```

```
                        if ( SQLBindCol(m_hstmt, (UWORD)(i+1),
SQL_C_SLONG, &m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )

                ThrowError(CODBCERR::eBindCol);
                }

                //Compose Delivery statement
                _snwprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryCommand[0]),
                        L"{call %stpcc_delivery (?,?)}",
m_szSPPrefix);
}

void CTPCC_ODBC::Delivery()
{
                RETCODE                 rc;
                int                             iTryCount = 0;

                m_hstmt = m_hstmtDelivery;

                while (TRUE)
                {
                        try
                        {
                                rc =
SQLExecDirectW(m_hstmt, m_szDeliveryCommand, SQL_NTS);
                                if (rc != SQL_SUCCESS && rc
!= SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

                                if ( SQLFetch(m_hstmt) ==
SQL_ERROR )

                ThrowError(CODBCERR::eFetch);

                                SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                m_txn.Delivery.exec_status_code = eOK;
                                break;
                        }
                        catch (CODBCERR *e)
                        {
                                if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                        throw;

                                // hit deadlock; backoff for
increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
                }
//              if (iTryCount)
//                      throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}
```

db_odbc.dll tpcc_odbc.h

```
/*              FILE:                   TPCC_ODBC.H
 *                                              Microsoft TPC-C
Kit Ver. 4.20.000
 *                                              Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                              Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:        Header file for TPC-C txn class
implementation.
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */
#pragma once

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

#define iMAX_SP_NAME_LEN 256        //maximum length of a stored
procedure name with parameters

class CODBCERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eAllocConn,
        // error from SQLAllocConnect
                        eAllocHandle,           //
error from SQLAllocHandle               eConnOption,
        // error from SQLSetConnectOption
                        eConnect,
        // error from SQLConnect
                        eAllocStmt,
        // error from SQLAllocStmt
                        eExecDirect,            //
error from SQLExecDirect
                        eBindParam,
        // error from SQLBindParameter
                        eBindCol,
        // error from SQLBindCol
                        eFetch,
                // error from SQLFetch
                        eFetchScroll,           //
error from SQLFetchScroll
                        eMoreResults,
        // error from SQLMoreResults
                        ePrepare,
        // error from SQLPrepare
                        eExecute,
        // error from SQLExecute
                        eSetEnvAttr,            //
error from SQLSetEnvAttr
                        eSetStmtAttr            //
error from SQLSetStmtAttr
                };

                CODBCERR(void)
                {
                        m_eAction = eNone;
                        m_NativeError = 0;
                        m_bDeadLock = FALSE;
                        m_odbcerrstr = NULL;
                };

                ~CODBCERR()
                {
                        if (m_odbcerrstr != NULL)
                                delete []
m_odbcerrstr;
                };

                ACTION      m_eAction;
                int                     m_NativeError;
                BOOL        m_bDeadLock;
                char   *m_odbcerrstr;

                int                     ErrorType()
{return ERR_TYPE_ODBC;};
                char*       ErrorTypeStr() { return
"ODBC"; }
```

```
                int                     ErrorNum()
{return m_NativeError;};
                char*           ErrorText() {return
m_odbcerrstr;};
                int                     ErrorAction() {
return (int)m_eAction; }
};

class CTPCC_ODBC_ERR : public CBaseErr
{
        public:
                enum TPCC_ODBC_ERRS
                {
                        ERR_WRONG_SP_VERSION =
1,      // "Wrong version of stored procs on database server"
                        ERR_INVALID_CUST,
                // "Invalid Customer id,name."
                        ERR_NO_SUCH_ORDER,
                // "No orders found for customer."
                        ERR_RETRIED_TRANS,
                // "Retries before transaction succeeded."
                };

                CTPCC_ODBC_ERR( int iErr ) { m_errno =
iErr; m_iTryCount = 0; };

                CTPCC_ODBC_ERR( int iErr, int iTryCount ) {
m_errno = iErr; m_iTryCount = iTryCount; };

                int                     m_errno;
                int                     m_iTryCount;

                int                     ErrorType()
{return ERR_TYPE_TPCC_ODBC;};
                char*           ErrorTypeStr() { return "TPCC
ODBC"; }
                int                     ErrorNum()
{return m_errno;};

                char*           ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
        private:
                // declare variables and private functions
here...
                BOOL                    m_bDeadlock;
                // transaction was selected as deadlock
victim
                int
        m_MaxRetries;                   // retry count on
deadlock
                SQLHENV                 m_henv;
                                // ODBC environment handle
                SQLHDBC                 m_hdbc;
                SQLHSTMT        m_hstmt;
        // the current hstmt

                SQLHSTMT        m_hstmtNewOrder;
                SQLHSTMT
        m_hstmtNewOrderNoDuplicates;    // NewOrder with
one result set for lineitem details
                SQLHSTMT        m_hstmtPayment;
                SQLHSTMT        m_hstmtDelivery;
                SQLHSTMT        m_hstmtOrderStatus;
                SQLHSTMT        m_hstmtStockLevel;

                SQLHDESC        m_descNewOrderCols1;
                SQLHDESC        m_descNewOrderCols2;
                SQLHDESC
        m_descNewOrderNoDuplicatesCols1;        // NewOrder with
one result set for lineitem details
                SQLHDESC
        m_descNewOrderNoDuplicatesCols2;        // NewOrder with
one result set for lineitem details
                SQLHDESC        m_descOrderStatusCols1;
                SQLHDESC        m_descOrderStatusCols2;

                wchar_t
        m_szSPPrefix[32];               // stored procedures prefix

                wchar_t
        m_szNewOrderCommand[iMAX_SP_NAME_LEN];
                wchar_t
        m_szNewOrderNoDuplicatesCommand[iMAX_SP_NAME_LE
N];
                int
        m_iBeginNewOrderVariablePart;           // begining of the
variable part in NewOrder statement
                int
        m_iBeginNewOrderNoDuplicatesVariablePart;               //
begining of the variable part in NewOrder statement
                wchar_t
        m_szPaymentCommand[iMAX_SP_NAME_LEN];
                wchar_t
        m_szDeliveryCommand[iMAX_SP_NAME_LEN];
                wchar_t
        m_szOrderStatusCommand[iMAX_SP_NAME_LEN];
                wchar_t
        m_szStockLevelCommand[iMAX_SP_NAME_LEN];

                // new-order specific fields
                SQLUINTEGER             m_RowsFetched;
                SQLUINTEGER             m_BindOffset;
                int
        m_no_commit_flag;

                // tpcc_neworder_new flag
                BOOL
        m_bCallNoDuplicatesNewOrder;

                void ThrowError( CODBCERR::ACTION
eAction );

                void InitNewOrderParams();
                void InitPaymentParams();
                void InitDeliveryParams();
                void InitStockLevelParams();
                void InitOrderStatusParams();

                union
                {
                        NEW_ORDER_DATA
        NewOrder;
                        PAYMENT_DATA
        Payment;
                        DELIVERY_DATA
        Delivery;
                        STOCK_LEVEL_DATA
        StockLevel;
                        ORDER_STATUS_DATA
        OrderStatus;
                }
        m_txn;

                bool DuplicatesInNewOrder();
                void NewOrderDuplicates();
                void NewOrderNoDuplicates();

        public:
                CTPCC_ODBC(             LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword,

        LPCSTR szHost, LPCSTR szDatabase,

        LPCWSTR szSPPrefix, BOOL bCallNoDuplicatesNewOrder);
                ~CTPCC_ODBC(void);

                inline PNEW_ORDER_DATA
        BuffAddr_NewOrder()             { return
&m_txn.NewOrder;    };
                inline PPAYMENT_DATA
        BuffAddr_Payment()              { return
&m_txn.Payment;     };
```

```
                        inline PDELIVERY_DATA
        BuffAddr_Delivery()             { return
&m_txn.Delivery;    };
                        inline PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()       { return &m_txn.StockLevel;
};
                        inline PORDER_STATUS_DATA
        BuffAddr_OrderStatus()      { return &m_txn.OrderStatus;
};

                        void NewOrder                   ();
                        void Payment            ();
                        void Delivery           ();
                        void StockLevel                 ();
                        void OrderStatus        ();

};


// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
        (               LPCSTR szServer, LPCSTR szUser, LPCSTR
szPassword,
                        LPCSTR szHost, LPCSTR szDatabase,
                        LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCWSTR, BOOL);
```

tm_com_dll\src\tpcc_com.h

```
/*      FILE:                   TPCC_COM.H
 *                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                      Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                      not yet audited
 *
 *      PURPOSE:    Header file for TPC-C COM+ class
implementation.
 *
 *  Change history:
 *                  4.20.000 - first version
 */
```

```
#pragma once
#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
        private:
                char m_szErrorText[64];

        public:
                // use this interface for genuine COM errors
                CCOMERR( HRESULT hr )
                {
                        m_hr = hr;
                        m_iErrorType = 0;
                        m_iError = 0;
                }

                // use this interface to impersonate a non-
COM error type
                CCOMERR( int iErrorType, int iError )
                {
                        m_iErrorType = iErrorType;
                        m_iError = iError;
                        m_hr = S_OK;
                }

                int             m_hr;
                int             m_iErrorType;
                int             m_iError;

                // A CCOMERR class can impersonate
another class, which happens if the error
                // was not actually a COM Services error, but
was simply transmitted back via COM.
                int ErrorType()
                {
                        if (m_iErrorType == 0)
                                return
ERR_TYPE_COM;
                        else
                                return
m_iErrorType;
                }

                char *ErrorTypeStr() { return "COM"; }
                int ErrorNum() {return m_hr;}

                char *ErrorText()
                {
                        if (m_hr == S_OK)
                                sprintf(
m_szErrorText, "Error: Class %d, error # %d", m_iErrorType, m_iError
);
                        else
                                sprintf(
m_szErrorText, "Error: COM HRESULT %x", m_hr );
                        return m_szErrorText;
                }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
        private:
                BOOL m_bSinglePool;

                // COM Interface pointers
                ITPCC*
        m_pNewOrder;
                ITPCC*
        m_pPayment;
                ITPCC*
        m_pStockLevel;
                ITPCC*
        m_pOrderStatus;

                struct COM_DATA
                {
                        int ErrorType;
                        int error;
                        union
                        {

        NEW_ORDER_DATA          NewOrder;
                                PAYMENT_DATA
                Payment;
                                DELIVERY_DATA
                Delivery;

        STOCK_LEVEL_DATA        StockLevel;

        ORDER_STATUS_DATA       OrderStatus;
                        } u;
                } *m_pTxn;

                VARIANT m_vTxn;
        public:
                CTPCC_COM(BOOL bSinglePool);
                ~CTPCC_COM(void);
```

```
                        inline PNEW_ORDER_DATA
        BuffAddr_NewOrder()             { return
&m_pTxn->u.NewOrder;    };
                        inline PPAYMENT_DATA
        BuffAddr_Payment()              { return
&m_pTxn->u.Payment;     };
                        inline PDELIVERY_DATA
        BuffAddr_Delivery()             { return
&m_pTxn->u.Delivery;    };
                        inline PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()   { return &m_pTxn-
>u.StockLevel;  };
                        inline PORDER_STATUS_DATA
        BuffAddr_OrderStatus()  { return &m_pTxn-
>u.OrderStatus; };

                        void NewOrder                   ();
                        void Payment            ();
                        void StockLevel                 ();
                        void OrderStatus        ();
                        void Delivery           () { throw new
CCOMERR(E_NOTIMPL); }  // not supported
};


inline void ReleaseInterface(IUnknown *pUnk)
{
        if (pUnk)
        {
                pUnk->Release();
                pUnk = NULL;
        }
}


// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

db_oledb_dll\tpcc_oledb.h

```
/*      FILE:                   TPCC_OLEDB.H
 *                                              Microsoft TPC-C
Kit Ver. 4.20.000                               Copyright
Microsoft, 1999-2004
 *                                              Written by Sergey
Vasilevskiy
 *      All Rights Reserved
 *
 *
 *
 *      PURPOSE:    Header file for TPC-C txn class OLE DB
implementation.
 *
 *
 */
#pragma once

// need to declare functions for import, unless define has already been
created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

#define iMAX_SP_NAME_LEN 256        //maximum length of a stored
procedure name with parameters

// Type of parameter and result set column bindings.
enum eBindingType
{
        eInputParameter,
        eOutputParameter,
        eInputOutputParameter,
        eOutputColumn
};

class COLEDBERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eQueryInterface,
                // error from QueryInterface
                        eCreateSession,
                        eCreateCommand,
                        eSetCommandText,
                        eExecute,
                // = 6
                        eCreateAccessor,
                        ePrepare,
                        eGetNextRows,
                        eGetData,
                        eGetResult
                // = 11 };

                COLEDBERR(LPCTSTR szLoc)
                        : CBaseErr(szLoc)
                {
                        m_eAction = eNone;
                        m_NativeError = 0;
                        m_bDeadLock = FALSE;
                        m_OLEDBErrStr = NULL;
                };

                ~COLEDBERR()
                {
                        if (m_OLEDBErrStr != NULL)
                                delete []
m_OLEDBErrStr;
                };

                ACTION  m_eAction;
                int                     m_NativeError;
                BOOL    m_bDeadLock;
                char   *m_OLEDBErrStr;

                int                     ErrorType()
{return ERR_TYPE_OLEDB;};
                char*   ErrorTypeStr() { return
"OLEDB"; }
                int                     ErrorNum()
{return m_NativeError;};
                char*   ErrorText() {return
m_OLEDBErrStr;};
                int                     ErrorAction() {
return (int)m_eAction; }

};

class CTPCC_OLEDB_ERR : public CBaseErr
{
        public:
                enum TPCC_OLEDB_ERRS
                {
                        ERR_WRONG_SP_VERSION =
1,      // "Wrong version of stored procs on database server"
                        ERR_INVALID_CUST,
                // "Invalid Customer id,name."
                        ERR_NO_SUCH_ORDER,
                // "No orders found for customer."
                        ERR_RETRIED_TRANS,
                // "Retries before transaction succeeded."
                };
```

```
                CTPCC_OLEDB_ERR( int iErr ) { m_errno =
iErr; m_iTryCount = 0; };

                CTPCC_OLEDB_ERR( int iErr, int iTryCount )
{ m_errno = iErr; m_iTryCount = iTryCount; };

                int             m_errno;
                int             m_iTryCount;

                int             ErrorType()
{return ERR_TYPE_TPCC_OLEDB;};
                char*   ErrorTypeStr() { return "TPCC
OLEDB"; }
                int             ErrorNum()
{return m_errno;};

                char*   ErrorText();
};

class DllDecl CTPCC_OLEDB : public CTPCC_BASE
{
        private:
                // declare variables and private functions
here...
                BOOL
                m_bDeadlock;                    //
transaction was selected as deadlock victim
                int
                        m_MaxRetries;
        // retry count on deadlock

                DBPROPSET
        m_rgInitPropSet;                // initialization
property set used to establish a connection
                DBPROP
                m_InitProperties[4];    // individual
initialization properties

                IDBCreateSession*
        m_pIDBCreateSession;    // session (connection)
interface
                IDBCreateCommand*
        m_pIDBCreateCommand;  // SQL command creation
interface

                IMalloc*
        m_pIMalloc;                     //
Needed to release error strings.

                // StockLevel
                ICommandText*
        m_pIStockLevelCommand;
                HACCESSOR
        m_hStockLevelInputAccessor;     // accessor to
bind input parameters   HACCESSOR
        m_hStockLevelOutputAccessor;    // accessor to
bind output columns
                DBPARAMS
        m_StockLevelExecuteParams;              //
parameter structure for Execute

                // NewOrder
                // One prepared command for each possible
number of new order line items
                ICommandText*
        m_pINewOrderCommand[MAX_OL_NEW_ORDER_ITEMS];
                // accessors to bind input parameters
                // one for each possible number of new
order line items
                HACCESSOR
        m_hNewOrderInputAccessor[MAX_OL_NEW_ORDER_ITEM
S];
                // accessor to bind output columns of the
first rowset
                HACCESSOR
        m_hNewOrderOutputAccessor[MAX_OL_NEW_ORDER_ITE
MS];
                // accessor to bind output columns of the
second rowset
                HACCESSOR
        m_hNewOrderOutputAccessor2[MAX_OL_NEW_ORDER_IT
EMS];
                // parameter structure for Execute
                DBPARAMS
        m_NewOrderExecuteParams[MAX_OL_NEW_ORDER_ITEM
S];

                // Payment
                ICommandText*
        m_pIPaymentCommand;
                HACCESSOR
        m_hPaymentInputAccessor;        // accessor to
bind input parameters
                HACCESSOR
        m_hPaymentOutputAccessor;       // accessor to
bind output columns
                DBPARAMS
        m_PaymentExecuteParams;                 //
parameter structure for Execute

                // OrderStatus
                ICommandText*
        m_pIOrderStatusCommand;
                HACCESSOR
        m_hOrderStatusInputAccessor;    // accessor to
bind input parameters   HACCESSOR
        m_hOrderStatusOutputAccessor;   // accessor to
bind output columns
                HACCESSOR
        m_hOrderStatusOutputAccessor2;  // accessor to
bind output columns
                DBPARAMS
        m_OrderStatusExecuteParams;             //
parameter structure for Execute

                // Delivery
                ICommandText*
        m_pIDeliveryCommand;
                HACCESSOR
        m_hDeliveryInputAccessor;       // accessor to
bind input parameters
                HACCESSOR
        m_hDeliveryOutputAccessor;      // accessor to
bind output columns
                DBPARAMS
        m_DeliveryExecuteParams;// parameter structure for
Execute

                wchar_t
                m_szSPPrefix[32];       // stored
procedures prefix

                // new-order specific fields
                int
                        m_no_commit_flag;

                void ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR szLocation
);

                void CheckSPVersion();

                void InitNewOrderParams();
                void InitPaymentParams();
                void InitDeliveryParams();
                void InitStockLevelParams();
                void InitOrderStatusParams();

                // Helper function to create and prepare a
command
                void CreateCommand(wchar_t*
szSQLCommand, ICommandText** ppICommandText);
                // Helper function to prepare a command
                void PrepareCommand(ICommandText*
pICommand);
```

```
                        // Helper function to fill one binding
                        // Used for both input parameter and output
column bindings
                        void SetBinding(DBBINDING* pDBBinding,
size_t obValue, size_t cbMaxLen, DBTYPE wType);

                        // Helper function to initialize an array of
bindings
                        void InitBindings(DBBINDING* pDBBindings,
int iCount, eBindingType BindingType);

                        union
                        {
                                NEW_ORDER_DATA
        NewOrder;
                                PAYMENT_DATA
        Payment;
                                DELIVERY_DATA
        Delivery;
                                STOCK_LEVEL_DATA
        StockLevel;
                                ORDER_STATUS_DATA
        OrderStatus;
                        }
        m_txn;

        public:
                        CTPCC_OLEDB(LPCSTR szServer, LPCSTR
szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase,
LPCWSTR szSPPrefix);
                        ~CTPCC_OLEDB(void);

                        inline PNEW_ORDER_DATA
        BuffAddr_NewOrder()             { return
&m_txn.NewOrder;    };
                        inline PPAYMENT_DATA
        BuffAddr_Payment()              { return
&m_txn.Payment;     };
                        inline PDELIVERY_DATA
        BuffAddr_Delivery()             { return
&m_txn.Delivery;    };
                        inline PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()   { return &m_txn.StockLevel;
};
                        inline PORDER_STATUS_DATA
        BuffAddr_OrderStatus()  { return &m_txn.OrderStatus;
};

                        void NewOrder                   ();
                        void Payment            ();
                        void Delivery           ();
                        void StockLevel                 ();
                        void OrderStatus        ();
};


// wrapper routine for class constructor
extern "C" DllDecl CTPCC_OLEDB* CTPCC_OLEDB_new
                ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase, LPCWSTR szSPPrefix );

typedef CTPCC_OLEDB* (TYPE_CTPCC_OLEDB)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCWSTR);
```

db_oledb_dll\src\tpcc_oledb.cpp

```
/*              FILE:                   TPCC_OLEDB.CPP
 *                                              Microsoft TPC-C
Kit Ver. 4.42.000
 *                                              Copyright
Microsoft, 2004
 *                                              Written by Sergey
Vasilevskiy
 *              All Rights Reserved
 *
 *
 *
 *              PURPOSE:    Implements OLEDB calls for TPC-C txns.
 *              Contact:    Charles Levine (clevine@microsoft.com)
 *
 *
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>
#include <stddef.h>

#define DBINITCONSTANTS
#include <oledb.h>
//#include <sqloledb.h> // Use MDAC
#include <sqlncli.h>        // Use SNAC
#include <oledberr.h>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_oledb.h"

#ifndef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

// version string; must match return value from tpcc_version stored
proc
const char    sVersion[] = "4.20.000";

const iMaxRetries = 10;                 // how many retries on
deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout expired";

// this needs to be the same as the max length of
machine/database/user/password in Benchcraft (engstut.h)
const static int            iMaxNameLen = 32;

BOOL APIENTRY DllMain(HMODULE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved)
{
        switch( ul_reason_for_call )
        {
                case DLL_PROCESS_ATTACH:

        DisableThreadLibraryCalls(hModule);

                        break;

                case DLL_PROCESS_DETACH:

                        break;

                default:
                        /* nothing */;
        }
        return TRUE;
}

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
 *
 */

char* CTPCC_OLEDB_ERR::ErrorText(void)
{
```

```
        int i;

        static SERRORMSG errorMsgs[] =
        {
                { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database server"   },
                { ERR_INVALID_CUST,
        "Invalid Customer id,name."
                                                            },
                { ERR_NO_SUCH_ORDER,                "No
orders found for customer."
                        },
                { ERR_RETRIED_TRANS,
        "Retries before transaction succeeded."
                },
                { 0,
                        ""



        }
        };

        static char szNotFound[] = "Unknown error number.";

        for(i=0; errorMsgs[i].szMsg[0]; i++)
        {
                if ( m_errno == errorMsgs[i].iError )
                        break;
        }
        if ( !errorMsgs[i].szMsg[0] )
                return szNotFound;
        else
                return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_OLEDB* CTPCC_OLEDB_new(
        LPCSTR szServer,                    // name of SQL
server
        LPCSTR szUser,                                  //
user name for login
        LPCSTR szPassword,                  // password for
login
        LPCSTR szHost,                                  //
not used
        LPCSTR szDatabase,                  // name of
database to use
        LPCWSTR szSPPrefix )                // prefix to
append to the stored procedure names
{
        return new CTPCC_OLEDB( szServer, szUser, szPassword,
szHost, szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
        LPCSTR szServer,
        // name of SQL server
        LPCSTR szUser,
                // user name for login
        LPCSTR szPassword,
        // password for login
        LPCSTR szHost,
                // not used
        LPCSTR szDatabase,
        // name of database to use
        LPCWSTR szSPPrefix
        // prefix to append to the stored procedure names
        )
        : m_pIMalloc(NULL)
{
        int
        iRc;
        int
        i;
        HRESULT                                         hr;

        IDBInitialize*          pIDBInitialize = NULL;
        // data source interface
        IDBProperties*                  pIDBProperties =
NULL;
        ICommandText*                   pICommandText;
                                        // SQL command
without parameters
        wchar_t
        szwServer[iMaxNameLen];         // Unicode string
used to convert to BSTR
        wchar_t
        szwDatabase[iMaxNameLen];       // Unicode string
used to convert to BSTR
        wchar_t
        szwUser[iMaxNameLen];           // Unicode string
used to convert to BSTR
        wchar_t
        szwPassword[iMaxNameLen];       // Unicode string
used to convert to BSTR

        // Copy stored procedures prefix
        wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

        // Convert single byte ANSI strings to Unicode (for later
conversion to BSTR)
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szServer, (int)strlen(szServer)+1, szwServer,
iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szDatabase, (int)strlen(szDatabase)+1,
szwDatabase, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szUser, (int)strlen(szUser)+1, szwUser,
iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szPassword, (int)strlen(szPassword)+1,
szwPassword, iMaxNameLen);

        // Initialize COM library to be able to use OLE-DB
interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB component
        //hr = CoCreateInstance(CLSID_SQLOLEDB, // GUID of
SQLOLEDB component
        // Compile for SNAC
        hr = CoCreateInstance(CLSID_SQLNCLI,// GUID of
SQLNCLI component
                NULL,                                   //
not defining an aggregate component, so NULL
                CLSCTX_INPROC_SERVER,           // run the
component in our process
                IID_IDBInitialize,
                (void **) &pIDBInitialize);
   /*
   Initialize the property values needed
   to establish the connection.
   */
   for(i = 0; i < 4; i++)
      VariantInit(&m_InitProperties[i].vValue);
   //Server name.
   m_InitProperties[0].dwPropertyID  = DBPROP_INIT_DATASOURCE;
   m_InitProperties[0].vValue.vt     = VT_BSTR;
   m_InitProperties[0].vValue.bstrVal= SysAllocString(szwServer);
   m_InitProperties[0].dwOptions     = DBPROPOPTIONS_REQUIRED;
   m_InitProperties[0].colid         = DB_NULLID;
   //Database.
   m_InitProperties[1].dwPropertyID  = DBPROP_INIT_CATALOG;
   m_InitProperties[1].vValue.vt     = VT_BSTR;
   m_InitProperties[1].vValue.bstrVal= SysAllocString(szwDatabase);
   m_InitProperties[1].dwOptions     = DBPROPOPTIONS_REQUIRED;
   m_InitProperties[1].colid         = DB_NULLID;
   //Username (login).
   m_InitProperties[2].dwPropertyID  = DBPROP_AUTH_USERID;
   m_InitProperties[2].vValue.vt     = VT_BSTR;
   m_InitProperties[2].vValue.bstrVal= SysAllocString(szwUser);
   m_InitProperties[2].dwOptions     = DBPROPOPTIONS_REQUIRED;
   m_InitProperties[2].colid         = DB_NULLID;
   //Password.
   m_InitProperties[3].dwPropertyID  = DBPROP_AUTH_PASSWORD;
   m_InitProperties[3].vValue.vt     = VT_BSTR;
   m_InitProperties[3].vValue.bstrVal= SysAllocString(szwPassword);
```

```
    m_InitProperties[3].dwOptions     = DBPROPOPTIONS_REQUIRED;
    m_InitProperties[3].colid         = DB_NULLID;
    /*
    Construct the DBPROPSET structure(m_rgInitPropSet). The
    DBPROPSET structure is used to pass an array of DBPROP
    structures (m_InitProperties) to the SetProperties method.
    */
    m_rgInitPropSet.guidPropertySet = DBPROPSET_DBINIT;
    m_rgInitPropSet.cProperties    = 4;
    m_rgInitPropSet.rgProperties   = m_InitProperties;
    //Set initialization properties.
                if (FAILED(hr = pIDBInitialize-
>QueryInterface(IID_IDBProperties,


                                (void **)&pIDBProperties)))
                {
                                ThrowError(pIDBInitialize,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB()");
                }

    hr = pIDBProperties->SetProperties(1, &m_rgInitPropSet);

    pIDBProperties->Release();
    //Now establish the connection to the data source.
    hr = pIDBInitialize->Initialize();

                // Free BSTR property strings
                for(i = 0; i < 4; i++)
                {
      SysFreeString(m_InitProperties[i].vValue.bstrVal);
                }

                hr = pIDBInitialize-
>QueryInterface(IID_IDBCreateSession, (void
**)&m_pIDBCreateSession);

                // Releasing this has no effect on the SQL Server
connection
    // of the data source object because of the reference maintained by
    // m_pIDBCreateSession.
    pIDBInitialize->Release();
    pIDBInitialize = NULL;

                hr = m_pIDBCreateSession->CreateSession(NULL,
IID_IDBCreateCommand, (IUnknown **)&m_pIDBCreateCommand);
                if (FAILED(hr))
                {
                                ThrowError(m_pIDBCreateSession,
COLEDBERR::eCreateSession, "CTPCC_OLEDB()");
                }

                hr = m_pIDBCreateCommand->CreateCommand(NULL,
IID_ICommandText, (IUnknown **)&pICommandText);
                if (FAILED(hr))
                {
                                ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CTPCC_OLEDB()");
                }

                hr = pICommandText->SetCommandText(DBGUID_SQL,
L"set nocount on set XACT_ABORT ON");
                if (FAILED(hr))
                {
                                ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CTPCC_OLEDB()");
                }

                hr = pICommandText->Execute(NULL, IID_NULL, NULL,
NULL, NULL);
                if (FAILED(hr))
                {
                                ThrowError(pICommandText,
COLEDBERR::eExecute, "CTPCC_OLEDB()");
                }

                pICommandText->Release();

                // verify that version of stored procs on server is correct
                CheckSPVersion();

                // Get IMalloc interface
                hr = CoGetMalloc(1, (LPMALLOC *)&m_pIMalloc);

                // Bind parameters for each of the transactions
                InitNewOrderParams();
                InitPaymentParams();
                InitOrderStatusParams();
                InitDeliveryParams();
                InitStockLevelParams();
}

CTPCC_OLEDB::~CTPCC_OLEDB( void )
{
                if (m_pIMalloc != NULL)
                {
                                m_pIMalloc->Release();
                }
                m_pIPaymentCommand->Release();
                m_pIDBCreateCommand->Release();
                m_pIDBCreateSession->Release();

                CoUninitialize();               // uninitialize COM library
}

/*
*/              Check stored procedures version on the server.
void CTPCC_OLEDB::CheckSPVersion()
{
                HRESULT                                 hr;
                char
                db_sp_version[10];
                ICommandText*           pICommandText;
                IAccessor*              pIAccessor;
                IRowset*                pRowset;
                const ULONG             nOutputParams = 1;      //
output 1st result set columns
                HACCESSOR               hTpccVersionOutputAccessor;
                // Structure to bind in accessor
                DBBINDING
                acOutputDBBinding[nOutputParams];
                DBBINDSTATUS
                acOutputDBBindStatus[nOutputParams];
                LONG                                    cRows = 1;  //
number of rows returned in the rowset
                ULONG                                   cRowsObtained;
                HROW                                    rghRow;
                //returned row handles
                HROW*                                   prghRow =
&rghRow;


                hr = m_pIDBCreateCommand->CreateCommand(NULL,
IID_ICommandText, (IUnknown **)&pICommandText);
                if (FAILED(hr))
                {
                                ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CheckSPVersion()");
                }

                hr = pICommandText->SetCommandText(DBGUID_SQL,
L"{call tpcc_version}");
                if (FAILED(hr))
                {
                                ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CheckSPVersion()");
                }

                hr = pICommandText->QueryInterface(IID_IAccessor,
(void **)&pIAccessor);
                if (FAILED(hr))
                {
                                ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CheckSPVersion()");
                }

                // Now fill the binding information for result set 1 output
columns
```

```
		InitBindings(&acOutputDBBinding[0], nOutputParams,
eOutputColumn);

		// Binding for a rowset
		SetBinding(&acOutputDBBinding[0], 0,
sizeof(db_sp_version), DBTYPE_STR);

		hr = pIAccessor->CreateAccessor(
			DBACCESSOR_ROWDATA,
			nOutputParams,
			acOutputDBBinding,
			sizeof(db_sp_version),
			&hTpccVersionOutputAccessor,
			acOutputDBBindStatus);
		if (FAILED(hr))
		{
				ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "CheckSPVersion()");
		}

		hr = pICommandText->Execute(NULL, IID_IRowset,
NULL, NULL, (IUnknown **)&pRowset);
		if (FAILED(hr))
		{
				ThrowError(pICommandText,
COLEDBERR::eExecute, "CheckSPVersion()");
		}

		// Fetch the result row handle(s)
		hr = pRowset->GetNextRows(DB_NULL_HCHAPTER, 0,
cRows, &cRowsObtained, &prghRow);
		if (FAILED(hr))
		{
				ThrowError(pICommandText,
COLEDBERR::eGetNextRows, "CheckSPVersion()");
		}

		// Fetch the actual row data by handle
		hr = pRowset->GetData(rghRow,
hTpccVersionOutputAccessor, &db_sp_version);
		if (FAILED(hr))
		{
				ThrowError(pICommandText,
COLEDBERR::eGetData, "CheckSPVersion()");
		}

		// Release row(s)
		hr = pRowset->Release();

		pICommandText->Release();

		// Check the retrieved version
		if (strcmp(db_sp_version,sVersion)
throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_WRONG_SP_VERSION
);
}

void CTPCC_OLEDB::ThrowError( IUnknown* pObjectWithError,
COLEDBERR::ACTION eAction, LPCTSTR szLocation)
{
		HRESULT
		hr;
		//char
		szState[6];
		char
		szMsg[SQL_MAX_MESSAGE_LENGTH];
		char
		szTmp[6*SQL_MAX_MESSAGE_LENGTH];
		COLEDBERR
		*pOLEDBErr;				// not allocated
until needed (maybe never)
		int
			iLen;
		// Interfaces
		IErrorInfo*		pIErrorInfoAll		= NULL;
		IErrorInfo*		pIErrorInfoRecord	= NULL;
		IErrorRecords*		pIErrorRecords		= NULL;
		ISupportErrorInfo*	pISupportErrorInfo	= NULL;
		ISQLServerErrorInfo*	pISQLServerErrorInfo	= NULL;
		ISQLErrorInfo*
		pISQLErrorInfo	= NULL;

		// Information used when cannot get custom error object
		ERRORINFO
		BasicErrorInfo;
		BSTR
		bstrDescription;
		// Number of error records.
		ULONG		nRecs;
		ULONG		nRec;

		// SQL Server error information from ISQLServerErrorInfo.
		SSERRORINFO*		pSSErrorInfo = NULL;
		OLECHAR*		pSSErrorStrings = NULL;

		assert(pObjectWithError != NULL);

		pOLEDBErr = new COLEDBERR(szLocation);

		pOLEDBErr->m_NativeError = 0;
		pOLEDBErr->m_eAction = eAction;
		pOLEDBErr->m_bDeadLock = FALSE;

		szTmp[0] = 0;

		// Only ask for error information if the interface supports
it.		// Note: SQLOLEDB provider supports error interface, so
this check is
		// for good style only.
		hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**) &pISupportErrorInfo);
		if (FAILED(hr))
		{
			_snprintf(szMsg, sizeof(szMsg),
"SupportErrorInfo interface not supported (hr=0x%X)", hr);
			pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
			strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
			throw pOLEDBErr;
		}
		/*if (FAILED(pISupportErrorInfo-
>InterfaceSupportsErrorInfo(IID_InterfaceWithError)))
		{
			_snprintf(szMsg, sizeof(szMsg),
"InterfaceWithError interface not supported");
			pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
			strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
			return;
		}*/

		// Do not test the return of GetErrorInfo. It can succeed
and return
		// a NULL pointer in pIErrorInfoAll. Simply test the pointer.
		GetErrorInfo(0, &pIErrorInfoAll);

		if (pIErrorInfoAll != NULL)
		{
			// Test to see if it's a valid OLE DB IErrorInfo
interface
			// exposing a list of records.
			if (SUCCEEDED(pIErrorInfoAll-
>QueryInterface(IID_IErrorRecords,	(void**) &pIErrorRecords)))
			{
				pIErrorRecords-
>GetRecordCount(&nRecs);

				// Within each record, retrieve
information from each
				// of the defined interfaces.
				for (nRec = 0; nRec < nRecs;
nRec++)
				{
					// Request the
generic SQL error interface.
					pIErrorRecords-
>GetCustomErrorObject(nRec,
```

```
            IID_ISQLErrorInfo, // generic SQL error interface
            (IUnknown**) &pISQLErrorInfo);
                                                if (pISQLErrorInfo
!= NULL)
                                                {
                                                        //
Request SQL Server-specific error interface, not the generic SQL error
interface.
            pISQLErrorInfo->QueryInterface(
            IID_ISQLServerErrorInfo, // SQL Server error interface

            (void**) &pISQLServerErrorInfo);
                                                }
                                                // Test to ensure
the reference is valid, then
                                                // get error
information from ISQLServerErrorInfo.
                                                if
(pISQLServerErrorInfo != NULL)
                                                {
            pISQLServerErrorInfo->GetErrorInfo(&pSSErrorInfo,
&pSSErrorStrings);
                                                        //
ISQLServerErrorInfo::GetErrorInfo succeeds
                                                        //
even when it has nothing to return. Test the
                                                        //
pointers before using.
                                                        if
(pSSErrorInfo)
                                                        {
            // First, add the error message.

            // Convert Unicode error string to ANSI.
            WideCharToMultiByte(CP_THREAD_ACP, 0,
                        pSSErrorInfo->pwszMessage, -1,
                        szMsg, sizeof(szMsg),
                        NULL, NULL);
            // quit if there isn't enough room to concatenate error text
            if ( (strlen(szMsg) + 2) > (sizeof(szTmp) - strlen(szTmp))
)
                        break;

            // include line break after first error msg
            if (szTmp[0] != 0)
                        strcat( szTmp, "\r\n");
            // concatenate the error record to the overall error
message
            strcat( szTmp, szMsg );

            // Second, add the stored procedure name and line
number, if available.

            if (wcslen(pSSErrorInfo->pwszProcedure)>0)
            {
                        // Prefix with a line break
                        iLen = sprintf(szMsg, "\r\nProcedure: ");

                        // Convert Unicode error string to ANSI.
                        WideCharToMultiByte(CP_THREAD_ACP, 0,
                                    pSSErrorInfo->pwszProcedure,
-1,
                                    &szMsg[iLen], sizeof(szMsg) -
iLen,
                                    NULL, NULL);

                        // Check if have space to add the line
number.
                        // Assume the line number takes no more
than 3 digits.
                        if ((strlen(szMsg) + 4)< sizeof(szMsg))
                        {

                _snprintf(&szMsg[strlen(szMsg)], sizeof(szMsg),
                                                ":%d",
pSSErrorInfo->wLineNumber);
                        }

                        // quit if there isn't enough room to
concatenate error text
                        if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
                                    break;

                        // concatenate the error record to the overall
error message
                        strcat( szTmp, szMsg );

                        // copy the overall error string to the
exception
                        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szTmp)+1];
                        strcpy(pOLEDBErr->m_OLEDBErrStr,
szTmp);
            }

            // Third, capture the (first) database error

            if (pOLEDBErr->m_NativeError == 0 && pSSErrorInfo-
>lNative != 0)
            {
                        pOLEDBErr->m_NativeError = pSSErrorInfo-
>lNative;
```

```
                        // Check for deadlock error code and set the
deadlock flag
                        if (pSSErrorInfo->lNative == 1205)
                        {
                                pOLEDBErr->m_bDeadLock =
TRUE;
                        }
                }

                // IMalloc::Free needed to release references
                // on returned values.
                if (m_pIMalloc != NULL)
                {
                        m_pIMalloc->Free(pSSErrorStrings);
                        m_pIMalloc->Free(pSSErrorInfo);
                }
                                                        }

                pISQLServerErrorInfo->Release();
                                                }
                                                else
                                                {
                                                        //
Custom error object is not supported.
                                                        //
Use general OLE-DB error interface.
                                                        //
Get the numeric error code
                        pIErrorRecords->GetBasicErrorInfo(nRec,
&BasicErrorInfo);
                                                        if
(pOLEDBErr->m_NativeError == 0)
                                                        {
                        // Get the failed call HRESULT code, which is not really the
native error
                        pOLEDBErr->m_NativeError = BasicErrorInfo.hrError;
                                                        }

                                                        //
Try to get the string description of the error.
                        pIErrorRecords->GetErrorInfo(nRec,
LOCALE_USER_DEFAULT, (IErrorInfo**)&pIErrorInfoRecord);
                                                        if
(pIErrorInfoRecord)
                                                        {
                        pIErrorInfoRecord->GetDescription(&bstrDescription);

                        // Convert Unicode error string to ANSI.
                        WideCharToMultiByte(CP_THREAD_ACP, 0,
                                bstrDescription, -1,
                                szMsg, sizeof(szMsg),
                                NULL, NULL);

                        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
                        strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
                                                        }
                                                }
                                        }               // for()
                                }               // if
(SUCCEEDED(pIErrorInfoAll->QueryInterface(IID_IErrorRecords,
(void**) &pIErrorRecords)))
                                else
                                {               // No IErrorRecords interface
supported. Use default IErrorInfo.
                                        // Note: SQLOLEDB supports
IErrorRecords, so this check is for good style only.
                                        _snprintf(szMsg, sizeof(szMsg),
"IErrorRecords interface not supported");
                                        pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szMsg)+1];
                                        strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
                                }
                        pIErrorInfoAll->Release();

                }               // if (pIErrorInfoAll != NULL)
                else
                {
                        // No IErrorInfo interface supported.
                        // Note: SQLOLEDB supports IErrorInfo, so
this check is for good style only.
                        _snprintf(szMsg, sizeof(szMsg), "IErrorInfo
interface not supported");
                        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
                        strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
                }

                throw pOLEDBErr;
}

/*
*
*               Create a new command object from the SQL text passed
in.
*
*/
void CTPCC_OLEDB::CreateCommand(wchar_t* szSQLCommand,
                        // I: SQL query for the command

                                        ICommandText**
ppICommandText          // O: returned command object

                                        )
{
                HRESULT                                                 hr;

                // Create a new command object
                hr = m_pIDBCreateCommand->CreateCommand(NULL,
IID_ICommandText, (IUnknown **)ppICommandText);
                if (FAILED(hr))
                {
                        ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CTPCC_OLEDB::CreateCommand");
                }

                // Set command text
                hr = (*ppICommandText)-
>SetCommandText(DBGUID_SQL, szSQLCommand);
                if (FAILED(hr))
                {
                        ThrowError(*ppICommandText,
COLEDBERR::eSetCommandText, "CTPCC_OLEDB::CreateCommand");
                }
```

```
	// Prepare the command
	PrepareCommand(*ppICommandText);
}

/*
*	QueryInterface and Prepare in one function for simplicity.
*	DEFERRED PREPARE property is set to off to prepare
immediatelly.
*/
void CTPCC_OLEDB::PrepareCommand(ICommandText*
pICommandText)
{
	HRESULT 	hr;
	ICommandPrepare*	pICommandPrepare;
	ICommandProperties*	pICommandProperties;
	DBPROPSET 	rowSetPropSet;
	DBPROP
	rowSetProp;

	// Set the deferred prepare property to false.
	rowSetProp.dwPropertyID = SSPROP_DEFERPREPARE;
	memset(&rowSetProp.vValue, 0,
sizeof(rowSetProp.vValue));
	rowSetProp.dwOptions = DBPROPOPTIONS_REQUIRED;
	rowSetProp.colid = DB_NULLID;

	rowSetPropSet.cProperties = 1;
	rowSetPropSet.guidPropertySet =
DBPROPSET_SQLSERVERROWSET;
	rowSetPropSet.rgProperties = &rowSetProp;

	// Query interface for setting properties
	hr = pICommandText-
>QueryInterface(IID_ICommandProperties, (void
**)&pICommandProperties);
	if (FAILED(hr))
	{
		ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB::PrepareCommand");
	}

	// Set the property set
	hr = pICommandProperties->SetProperties(1,
&rowSetPropSet);
	if (FAILED(hr))
	{
		ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB::PrepareCommand");
	}

	// Get interface for preparing commands
	hr = pICommandText-
>QueryInterface(IID_ICommandPrepare, (void
**)&pICommandPrepare);
	if (FAILED(hr))
	{
		ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB::PrepareCommand");
	}

	// Prepare Payment command
	hr = pICommandPrepare->Prepare(0xFFFFFFFF);
	if (FAILED(hr))
	{
		ThrowError(pICommandPrepare,
COLEDBERR::ePrepare, "CTPCC_OLEDB::PrepareCommand");
	}
}

/*
*	Initialize fields of an array of bindings structures.
*	Needs to be called before setting individual
parameter/column bindings.
*/
void CTPCC_OLEDB::InitBindings(DBBINDING* pDBBindings,	//
IO: array of bindings

		int iCount,
		// I: number of elements in the array

		eBindingType BindingType)		// I:
what the bindings will be used for (parameters/columns)
{
	int i;

	for(i = 0; i < iCount; i++)
  {
		pDBBindings[i].iOrdinal = i + 1;
	pDBBindings[i].obLength = 0;
	pDBBindings[i].obStatus = 0;
	pDBBindings[i].pTypeInfo = NULL;
	pDBBindings[i].pObject = NULL;
	pDBBindings[i].pBindExt = NULL;
	pDBBindings[i].dwPart = DBPART_VALUE;

		switch (BindingType)
		{
		case eInputParameter:
			pDBBindings[i].eParamIO =
DBPARAMIO_INPUT;
			break;
		case eOutputParameter:
			pDBBindings[i].eParamIO =
DBPARAMIO_OUTPUT;
			break;
		case eInputOutputParameter:
			pDBBindings[i].eParamIO =
DBPARAMIO_INPUT | DBPARAMIO_OUTPUT;
			break;
		case eOutputColumn:
			pDBBindings[i].eParamIO =
DBPARAMIO_NOTPARAM;
			break;
		default:
			assert(false); // this should
never happen
		}

	pDBBindings[i].dwMemOwner = DBMEMOWNER_CLIENTOWNED;
	pDBBindings[i].dwFlags = 0;
		pDBBindings[i].bPrecision = 0;
	pDBBindings[i].bScale = 0;
  }
}

/*
*		Perform binding for one parameter or output column.
*
*/
void CTPCC_OLEDB::SetBinding(DBBINDING* pDBBinding, // I: binding
row structure

		size_t obValue,
		// I: parameter (column) offset in the user buffer

		size_t cbMaxLen,		// I:
parameter (column) length

		DBTYPE wType
		// I: parameter (column) type

		)
{
		pDBBinding->obValue = (ULONG)obValue;
		pDBBinding->cbMaxLen = (ULONG)cbMaxLen;
		pDBBinding->wType = wType;
}

void CTPCC_OLEDB::InitStockLevelParams()
{
		int
			i;
		HRESULT
		hr;
		wchar_t
		szName[iMAX_SP_NAME_LEN];
		IAccessor*
		pIAccessor;
```

```
const ULONG
        nInputParams = 3;       // input parameters
        const ULONG
        nOutputParams = 1;      // output 1st result set columns

        // Structure to bind in accessor
        DBBINDING
        acInputDBBinding[nInputParams];
        DBBINDSTATUS
        acInputDBBindStatus[nInputParams];
        DBBINDING
        acOutputDBBinding[nOutputParams];
        DBBINDSTATUS
        acOutputDBBindStatus[nOutputParams];

        // Set command text
        _snwprintf(szName, sizeof(szName)/sizeof(szName[0]),
                        L"{call %stpcc_stocklevel
(?,?,?)}", m_szSPPrefix);

        // Create and Prepare a new command object for
StockLevel.
        CreateCommand(szName, &m_pIStockLevelCommand);


        // Describe the consumer buffer by filling in the array
// of DBBINDING structures.  Each binding associates
// a single parameter to the consumer's buffer.
        InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

        i = 0;
// StockLevel parameter 1
        SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, w_id), sizeof(m_txn.StockLevel.w_id),
DBTYPE_I4);

// StockLevel parameter 2
        SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, d_id), sizeof(m_txn.StockLevel.d_id),
DBTYPE_UI1);

        // StockLevel parameter 3
        SetBinding(&acInputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, threshold),
sizeof(m_txn.StockLevel.threshold), DBTYPE_I2);

        hr = m_pIStockLevelCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
        if (FAILED(hr))
        {
                ThrowError(m_pIStockLevelCommand,
COLEDBERR::eQueryInterface, "InitStockLevelParams()");
        }

        hr = pIAccessor->CreateAccessor(
                DBACCESSOR_PARAMETERDATA,
                nInputParams,
                acInputDBBinding,
                sizeof(STOCK_LEVEL_DATA),
                &m_hStockLevelInputAccessor,
                acInputDBBindStatus);
        if (FAILED(hr))
        {
                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitStockLevelParams()");
        }

        m_StockLevelExecuteParams.cParamSets = 1;
        m_StockLevelExecuteParams.hAccessor =
m_hStockLevelInputAccessor;
        m_StockLevelExecuteParams.pData = &m_txn.StockLevel;

        // Now fill the binding information for result set 1 output
columns
        InitBindings(&acOutputDBBinding[0], nOutputParams,
eOutputColumn);

        // Binding for a rowset that may return more than one
row.
        i = 0;
        // StockLevel output column 1
        SetBinding(&acOutputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, low_stock),
sizeof(m_txn.StockLevel.low_stock), DBTYPE_I4);

        hr = pIAccessor->CreateAccessor(
                DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
                nOutputParams,
                acOutputDBBinding,
                sizeof(STOCK_LEVEL_DATA),
                &m_hStockLevelOutputAccessor,
                acOutputDBBindStatus);
        if (FAILED(hr))
        {
                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitStockLevelParams()");
        }
}

void CTPCC_OLEDB::StockLevel()
{
        HRESULT                 hr;
        int
        iRowset*                pRowset;
        iTryCount = 0;
        LONG                    cRows = 1;   //
number of rows returned in the rowset
        ULONG                   cRowsObtained;
        HROW                    rghRow;
        //returned row handles
        HROW*                   prghRow =
&rghRow;

        while (TRUE)
        {
                try
                {
                        // Execute the prepared
command
                        hr =
m_pIStockLevelCommand->Execute(NULL, IID_IRowset,
&m_StockLevelExecuteParams, NULL,


        (IUnknown **)&pRowset);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIStockLevelCommand,
COLEDBERR::eExecute, "StockLevel()");
                        }

                        // Fetch the result row
handle(s)
                        hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained,
&prghRow);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetNextRows, "StockLevel()");
                        }

                        // Fetch the actual row data by
handle
                        hr = pRowset-
>GetData(rghRow, m_hStockLevelOutputAccessor, &m_txn.StockLevel);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetData, "StockLevel()");
                        }

                        // Release row(s)
```

```
                                        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL, NULL);
                                        // Release rowset
                                        hr = pRowset->Release();


                m_txn.StockLevel.exec_status_code = eOK;

                                        break;
                        }
                        catch (COLEDBERR *e)
                        {
                                        if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                                throw;

                                        // hit deadlock; backoff for
increasingly longer period
                                        delete e;
                                        Sleep(10 * iTryCount);
                        }
                }

//              if (iTryCount)
//                      throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{
                int
                        i, j, iOlCount;
                HRESULT
                hr;
                wchar_t
                szName[iMAX_SP_NAME_LEN];
                IAccessor*
                pIAccessor;
    const ULONG
                nInputParams = 5 + 3*MAX_OL_NEW_ORDER_ITEMS;
                // input parameters
                const ULONG
                nOutputParams = 5;      // output 1st result set columns
                const ULONG
                nOutputParams2 = 8;     // output 2nd result set
columns
                // Structure to bind in accessor
                DBBINDING
                acInputDBBinding[nInputParams];
                DBBINDSTATUS
                acInputDBBindStatus[nInputParams];
                DBBINDING
                acOutputDBBinding[nOutputParams];
                DBBINDSTATUS
                acOutputDBBindStatus[nOutputParams];
                DBBINDING
                acOutputDBBinding2[nOutputParams2];
                DBBINDSTATUS
                acOutputDBBindStatus2[nOutputParams2];

                // Describe the consumer buffer by filling in the array
    // of DBBINDING structures.  Each binding associates
    // a single parameter to the consumer's buffer.
                InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

                i = 0;
    // NewOrder parameter 1
                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, w_id), sizeof(m_txn.NewOrder.w_id),
DBTYPE_I4);

    // NewOrder parameter 2
                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, d_id), sizeof(m_txn.NewOrder.d_id),
DBTYPE_UI1);

                // NewOrder parameter 3
                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, c_id), sizeof(m_txn.NewOrder.c_id),
DBTYPE_I4);

                // NewOrder parameter 4
                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

                // NewOrder parameter 5
                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_all_local),
sizeof(m_txn.NewOrder.o_all_local), DBTYPE_UI1);

                for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
                {
                                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_i_id),
sizeof(m_txn.NewOrder.OL[j].ol_i_id), DBTYPE_I4);
                                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_supply_w_id),
sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id), DBTYPE_I4);
                                SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_quantity),
sizeof(m_txn.NewOrder.OL[j].ol_quantity), DBTYPE_I2);

                }

                // Now fill the binding information for result set 1 output
columns
                InitBindings(&acOutputDBBinding[0], nOutputParams,
eOutputColumn);

                // Binding for the order line rowsets (each consist of one
row).
                // Bind to offsets of the OL_NEW_ORDER_DATA structure
instead of NEW_ORDER_DATA.
                // IRowset::GetData() will be passed individual array slots
OL[i] to fetch the data
                // from the row set.

                i = 0;
                // NewOrder output column 1
                SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name), DBTYPE_STR);

                // NewOrder output column 2
                SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock), DBTYPE_I2);

                // NewOrder output column 3
                SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_brand_generic),
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), DBTYPE_STR);

                // NewOrder output column 4
                SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price), DBTYPE_R8);

                // NewOrder output column 5
                SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_amount),
sizeof(m_txn.NewOrder.OL[0].ol_amount), DBTYPE_R8);



                // Now fill the binding information for result set 2 output
columns
                InitBindings(&acOutputDBBinding2[0], nOutputParams2,
eOutputColumn);

                i = 0;
                // NewOrder output column 1
                SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, w_tax), sizeof(m_txn.NewOrder.w_tax),
DBTYPE_R8);
```

```
		// NewOrder output column 2
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, d_tax), sizeof(m_txn.NewOrder.d_tax),
DBTYPE_R8);

		// NewOrder output column 3
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_id), sizeof(m_txn.NewOrder.o_id),
DBTYPE_I4);

		// NewOrder output column 4
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_last), sizeof(m_txn.NewOrder.c_last),
DBTYPE_STR);

		// NewOrder output column 5
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_discount),
sizeof(m_txn.NewOrder.c_discount), DBTYPE_R8);

		// NewOrder output column 6
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_credit),
sizeof(m_txn.NewOrder.c_credit), DBTYPE_STR);

		// NewOrder output column 7
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_entry_d),
sizeof(m_txn.NewOrder.o_entry_d), DBTYPE_DBTIMESTAMP);

		// NewOrder output column 8
		SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_commit_flag),
sizeof(m_txn.NewOrder.o_commit_flag), DBTYPE_I2);

		for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
		{
			// Set command text first

			// Print the fixed first portion of parameters
			i = _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),

		L"{call %stpcc_neworder (?,?,?,?,?", m_szSPPrefix);

			// Now print the variable portion depending
on the number of order line parameters
			for (iOlCount = 0; iOlCount <= j;
++iOlCount)
			{
				i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",?,?,?");
			}

			// Print the fixed end
			if (j != MAX_OL_NEW_ORDER_ITEMS - 1)
			{	// append 'default' for the
parameters that are not used
				i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",default)}");
			}
			else		// using all 15 order line
parameters
			{
				i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L")}");
			}

			// Create and Prepare a new command
object for NewOrder.
			CreateCommand(szName,
&m_pINewOrderCommand[j]);

			// Now create the input accessor for this
prepared command
			hr = m_pINewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
			if (FAILED(hr))
			{

		ThrowError(m_pINewOrderCommand[j],
COLEDBERR::eQueryInterface, "InitNewOrderParams()");
			}

			hr = pIAccessor->CreateAccessor(


		DBACCESSOR_PARAMETERDATA,

								5 +
3 * (j + 1),


		acInputDBBinding,


		sizeof(NEW_ORDER_DATA),


		&m_hNewOrderInputAccessor[j],


		acInputDBBindStatus);
			if (FAILED(hr))
			{
				ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
			}


			m_NewOrderExecuteParams[j].cParamSets
= 1;
			// m_NewOrderExecuteParams.hAccessor is
set dynamically at run-time
			// based on the number of new order items
for the particular transaction call.
			m_NewOrderExecuteParams[j].hAccessor =
m_hNewOrderInputAccessor[j];
			m_NewOrderExecuteParams[j].pData =
&m_txn.NewOrder;

			// Create accessor for the first rowset
			hr = pIAccessor->CreateAccessor(
				DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
				nOutputParams,
				acOutputDBBinding,

		sizeof(OL_NEW_ORDER_DATA),

		&m_hNewOrderOutputAccessor[j],
				acOutputDBBindStatus);
			if (FAILED(hr))
			{
				ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
			}

			// Create accessor for the second rowset
			hr = pIAccessor->CreateAccessor(
				DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
				nOutputParams2,
				acOutputDBBinding2,
				sizeof(NEW_ORDER_DATA),

		&m_hNewOrderOutputAccessor2[j],
				acOutputDBBindStatus2);
			if (FAILED(hr))
			{
				ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
			}

			pIAccessor->Release();
		}
}

void CTPCC_OLEDB::NewOrder()
```

```
{
	HRESULT 	hr;
	int
	iTryCount = 0;
	IMultipleResults*	pMultipleResults;
	IRowset*	pRowset;
	IRowset*	pRowset2;
	LONG
	cRows = 1;   // number of rows returned in the 1st rowset
	ULONG
	cRowsObtained;
	HROW
	rghRows;	//returned row handles for the
1st result set
	HROW*
	prghRows = &rghRows;
	LONG
	cRows2 = 1; // number of rows returned in the 2nd
rowset
	ULONG
	cRowsObtained2;
	HROW
	rghRows2;	//returned row handle for the
2nd result set
	HROW*
	prghRows2 = &rghRows2;
	int
	i;
	long
	lRowsAffected;	// the number of affected rows
for a rowset
	int
	iHandleIndex;	// index into the handle arrays
based on the orders count

	// check whether any order lines are for a remote
warehouse
	m_txn.NewOrder.o_all_local = 1;
	for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
	{
		if (m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
		{
			m_txn.NewOrder.o_all_local =
0;  // at least one remote warehouse
			break;
		}
	}

	iHandleIndex = m_txn.NewOrder.o_ol_cnt - 1;	//
for convenience

	while (TRUE)
		try
		{
			// Execute the prepared
command (according to the number of new orders)
			// Ask for IMultipleResults
because it returns 2 rowsets.
			hr =
m_pINewOrderCommand[iHandleIndex]->Execute(


	NULL, IID_IMultipleResults,


	&m_NewOrderExecuteParams[iHandleIndex],


	NULL,


	(IUnknown **)&pMultipleResults);
			if (FAILED(hr))
			{

	ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eExecute, "NewOrder()");
			}

			//////////////////////////
			// Get order line results
			//////////////////////////
			m_txn.NewOrder.total_amount
= 0;
			for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; ++i)
			{
				// Get the first
rowset object
				hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **)&pRowset);
				if (FAILED(hr))
				{
					char
szTmp[256];


	_snprintf(szTmp, sizeof(szTmp), "NewOrder() result set
%d, hr=0x%X", i, hr);


	ThrowError(m_pINewOrderCommand[m_txn.NewOrder.o_
ol_cnt - 1], COLEDBERR::eGetResult, szTmp);
				}
				// Fetch the result
row handle(s)
				hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained,
&prghRows);
				if (FAILED(hr))
				{

	ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetNextRows, "NewOrder()");
				}

				// Fetch the
actual row data by handle
				hr = pRowset-
>GetData(rghRows, m_hNewOrderOutputAccessor[iHandleIndex],
&m_txn.NewOrder.OL[i]);
				if (FAILED(hr))
				{

	ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetData, "NewOrder()");
				}


	m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;

				// Release row(s)
				hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL, NULL);
				// Release rowset
				hr = pRowset-
>Release();
			}

			//////////////////////////////
			// Get the second rowset
object
			//////////////////////////////
			hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset, &lRowsAffected, (IUnknown
**)&pRowset2);
			if (FAILED(hr))
			{
				char szTmp[256];

				_snprintf(szTmp,
sizeof(szTmp), "NewOrder() result set %d, hr=%d", i, hr);
```

```
		ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetResult, szTmp);
				}

				// Fetch the result row
handle(s)
				hr = pRowset2-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows2, &cRowsObtained2,
&prghRows2);
				if (FAILED(hr))
				{

		ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetNextRows, "NewOrder()");
				}

				// Fetch the actual row data by
handle
				hr = pRowset2-
>GetData(rghRows2, m_hNewOrderOutputAccessor2[iHandleIndex],
&m_txn.NewOrder);
				if (FAILED(hr))
				{

		ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetData, "NewOrder()");
				}

				// Release row(s)
				hr = pRowset2-
>ReleaseRows(cRowsObtained2, prghRows2, NULL, NULL, NULL);
				// Release rowset
				hr = pRowset2->Release();

				// Release the common
MultipleResults interface
				hr = pMultipleResults-
>Release();

				if (m_txn.NewOrder.o_all_local
== 1)
				{

		m_txn.NewOrder.total_amount *= ((1 +
m_txn.NewOrder.w_tax + m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

		m_txn.NewOrder.exec_status_code = eOK;
				}
				else
				{

		m_txn.NewOrder.exec_status_code = eInvalidItem;

				break;

			}
			catch (COLEDBERR *e)
			{
				if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
					throw;

				// hit deadlock; backoff for
increasingly longer period
				delete e;
				Sleep(10 * iTryCount);
			}
		}

//		if (iTryCount)
//			throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitPaymentParams()
{
		int
			i;
		HRESULT
		hr;
		wchar_t
		szName[iMAX_SP_NAME_LEN];
		IAccessor*
		pIAccessor;
  const ULONG
		nInputParams = 7;	// input parameters
		const ULONG
		nOutputParams = 27;	// output result set columns
		// Structure to bind in accessor
		DBBINDING
		acInputDBBinding[nInputParams];
		DBBINDSTATUS
		acInputDBBindStatus[nInputParams];
		DBBINDING
		acOutputDBBinding[nOutputParams];
		DBBINDSTATUS
		acOutputDBBindStatus[nOutputParams];

		// Set command text
		_snwprintf(szName, sizeof(szName)/sizeof(szName[0]),
L"{call %stpcc_payment(?,?,?,?,?,?,?)}", m_szSPPrefix);

		// Create and Prepare a new command object for
Payment.	CreateCommand(szName, &m_pIPaymentCommand);

		// Describe the consumer buffer by filling in the array
  // of DBBINDING structures.  Each binding associates
  // a single parameter to the consumer's buffer.
		InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

		i = 0;
  // Payment parameter 1
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, w_id), sizeof(m_txn.Payment.w_id),
DBTYPE_I4);

  // Payment parameter 2
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_w_id), sizeof(m_txn.Payment.c_w_id),
DBTYPE_I4);

		// Payment parameter 3
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, h_amount),
sizeof(m_txn.Payment.h_amount), DBTYPE_R8);

  // Payment parameter 4
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, d_id), sizeof(m_txn.Payment.d_id),
DBTYPE_UI1);

		// Payment parameter 5
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_d_id), sizeof(m_txn.Payment.c_d_id),
DBTYPE_UI1);

		// Payment parameter 6
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_id), sizeof(m_txn.Payment.c_id),
DBTYPE_I4);

		// Payment parameter 7
		SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last), sizeof(m_txn.Payment.c_last),
DBTYPE_STR);

		hr = m_pIPaymentCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
		if (FAILED(hr))
		{
			ThrowError(m_pIPaymentCommand,
COLEDBERR::eQueryInterface, "InitPaymentParams()");
		}
```

```
        hr = pIAccessor->CreateAccessor(
                DBACCESSOR_PARAMETERDATA,
                nInputParams,
                acInputDBBinding,
                sizeof(PAYMENT_DATA),
                &m_hPaymentInputAccessor,
                acInputDBBindStatus);
        if (FAILED(hr))
        {
                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams()");
        }

        m_PaymentExecuteParams.cParamSets = 1;
        m_PaymentExecuteParams.hAccessor =
m_hPaymentInputAccessor;
        m_PaymentExecuteParams.pData = &m_txn.Payment;

        // Now fill the binding information for output columns
        InitBindings(&acOutputDBBinding[0], nOutputParams,
eOutputColumn);

        i = 0;
        // Payment output column 1
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_id), sizeof(m_txn.Payment.c_id),
DBTYPE_I4);

        // Payment output column 2
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last), sizeof(m_txn.Payment.c_last),
DBTYPE_STR);

        // Payment output column 3
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, h_date), sizeof(m_txn.Payment.h_date),
DBTYPE_DBTIMESTAMP);

        // Payment output column 4
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_1),
sizeof(m_txn.Payment.w_street_1), DBTYPE_STR);

        // Payment output column 5
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_2),
sizeof(m_txn.Payment.w_street_2), DBTYPE_STR);

        // Payment output column 6
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_city), sizeof(m_txn.Payment.w_city),
DBTYPE_STR);

        // Payment output column 7
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_state), sizeof(m_txn.Payment.w_state),
DBTYPE_STR);

        // Payment output column 8
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_zip), sizeof(m_txn.Payment.w_zip),
DBTYPE_STR);

        // Payment output column 9
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

        // Payment output column 10
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

        // Payment output column 11
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city), sizeof(m_txn.Payment.d_city),
DBTYPE_STR);

        // Payment output column 12
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state), sizeof(m_txn.Payment.d_state),
DBTYPE_STR);

        // Payment output column 13
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip), sizeof(m_txn.Payment.d_zip),
DBTYPE_STR);

        // Payment output column 14
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_first), sizeof(m_txn.Payment.c_first),
DBTYPE_STR);

        // Payment output column 15
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_middle), sizeof(m_txn.Payment.c_middle),
DBTYPE_STR);

        // Payment output column 16
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

        // Payment output column 17
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);
        // Payment output column 18
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city), sizeof(m_txn.Payment.d_city),
DBTYPE_STR);

        // Payment output column 19
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state), sizeof(m_txn.Payment.d_state),
DBTYPE_STR);

        // Payment output column 20
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip), sizeof(m_txn.Payment.d_zip),
DBTYPE_STR);

        // Payment output column 21
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_phone), sizeof(m_txn.Payment.c_phone),
DBTYPE_STR);

        // Payment output column 22
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_since), sizeof(m_txn.Payment.c_since),
DBTYPE_DBTIMESTAMP);

        // Payment output column 23
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit), sizeof(m_txn.Payment.c_credit),
DBTYPE_STR);

        // Payment output column 24
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit_lim),
sizeof(m_txn.Payment.c_credit_lim), DBTYPE_R8);

        // Payment output column 25
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_discount),
sizeof(m_txn.Payment.c_discount), DBTYPE_R8);

        // Payment output column 26
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_balance),
sizeof(m_txn.Payment.c_balance), DBTYPE_R8);

        // Payment output column 27
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_data), sizeof(m_txn.Payment.c_data),
DBTYPE_STR);
```

```
        hr = pIAccessor->CreateAccessor(
                DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
                nOutputParams,
                acOutputDBBinding,
                sizeof(PAYMENT_DATA),
                &m_hPaymentOutputAccessor,
                acOutputDBBindStatus);
        if (FAILED(hr))
        {
                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams()");
        }
}

void CTPCC_OLEDB::Payment()
{
        HRESULT                         hr;
        int
        iTryCount = 0;
        IRowset*                pRowset;
        LONG                            cRows = 1;   //
number of rows returned in the rowset
        ULONG                           cRowsObtained;
        HROW                            rghRow;
        //returned row handles
        HROW*                           prghRow =
&rghRow;


        if (m_txn.Payment.c_id != 0)
                m_txn.Payment.c_last[0] = 0;

        while (TRUE)
        {
                try
                {
                        // Execute the prepared
command
                        hr = m_pIPaymentCommand-
>Execute(NULL, IID_IRowset, &m_PaymentExecuteParams, NULL,


        (IUnknown **)&pRowset);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIPaymentCommand,
COLEDBERR::eExecute, "Payment()");
                        }
```

```
                        // Fetch the result row
handle(s)
                        hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained,
&prghRow);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetNextRows, "Payment()");
                        }


                        // Fetch the actual row data by
handle
                        hr = pRowset-
>GetData(rghRow, m_hPaymentOutputAccessor, &m_txn.Payment);
                        if (FAILED(hr))
                        {

        ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetData, "Payment()");
                        }

                        // Release row(s)
                        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL, NULL);
                        // Release rowset
                        hr = pRowset->Release();


                        if (m_txn.Payment.c_id == 0)
                                throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
                        else

        m_txn.Payment.exec_status_code = eOK;

                        break;
                }
                catch (COLEDBERR *e)
                {
                        if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                                throw;

                        // hit deadlock; backoff for
increasingly longer period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }

//      if (iTryCount)
```

```
//                      throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitOrderStatusParams()
{
        int
                i;
        HRESULT
        hr;
        wchar_t
        szName[iMAX_SP_NAME_LEN];
        IAccessor*
        pIAccessor;
    const ULONG
        nInputParams = 4;       // input parameters
        const ULONG
        nOutputParams = 5;      // output 1st result set columns
        const ULONG
        nOutputParams2 = 8;     // output 2nd result set
columns
        // Structure to bind in accessor
        DBBINDING
        acInputDBBinding[nInputParams];
        DBBINDSTATUS
        acInputDBBindStatus[nInputParams];
        DBBINDING
        acOutputDBBinding[nOutputParams];
        DBBINDSTATUS
        acOutputDBBindStatus[nOutputParams];
        DBBINDING
        acOutputDBBinding2[nOutputParams2];
        DBBINDSTATUS
        acOutputDBBindStatus2[nOutputParams2];

        // Set command text
        _snwprintf(szName, sizeof(szName)/sizeof(szName[0]),
                        L"{call %stpcc_orderstatus
(?,?,?,?)}", m_szSPPrefix);

        // Create and Prepare a new command object for
OrderStatus.
        CreateCommand(szName, &m_pIOrderStatusCommand);


        // Describe the consumer buffer by filling in the array
    // of DBBINDING structures.  Each binding associates
    // a single parameter to the consumer's buffer.
        InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

        i = 0;
    // OrderStatus parameter 1
```

```
		SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, w_id), sizeof(m_txn.OrderStatus.w_id),
DBTYPE_I4);

   // OrderStatus parameter 2
		SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, d_id), sizeof(m_txn.OrderStatus.d_id),
DBTYPE_UI1);

		// OrderStatus parameter 3
		SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_id), sizeof(m_txn.OrderStatus.c_id),
DBTYPE_I4);

		// OrderStatus parameter 4
		SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

		hr = m_pIOrderStatusCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
		if (FAILED(hr))
		{
			ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eQueryInterface, "InitOrderStatusParams()");
		}

		hr = pIAccessor->CreateAccessor(
			DBACCESSOR_PARAMETERDATA,
			nInputParams,
			acInputDBBinding,
			sizeof(ORDER_STATUS_DATA),
			&m_hOrderStatusInputAccessor,
			acInputDBBindStatus);
		if (FAILED(hr))
		{
			ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
		}

		m_OrderStatusExecuteParams.cParamSets = 1;
		m_OrderStatusExecuteParams.hAccessor =
m_hOrderStatusInputAccessor;
		m_OrderStatusExecuteParams.pData =
&m_txn.OrderStatus;

		// Now fill the binding information for result set 1 output
columns
		InitBindings(&acOutputDBBinding[0], nOutputParams,
eOutputColumn);

		// Binding for a rowset that may return more than one
row.		// Bind to offsets of the OL_ORDER_STATUS_DATA
structure instead of ORDER_STATUS_DATA.
		// IRowset::GetData() will be passed individual array slots
OL[i] to fetch the data
		// from the row set.

		i = 0;
		// OrderStatus output column 1
		SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_supply_w_id),
sizeof(m_txn.OrderStatus.OL[0].ol_supply_w_id), DBTYPE_I4);

		// OrderStatus output column 2
		SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_txn.OrderStatus.OL[0].ol_i_id), DBTYPE_I4);

		// OrderStatus output column 3
		SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_quantity),
sizeof(m_txn.OrderStatus.OL[0].ol_quantity), DBTYPE_I2);

		// OrderStatus output column 4
		SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_amount),
sizeof(m_txn.OrderStatus.OL[0].ol_amount), DBTYPE_R8);

		// OrderStatus output column 5
		SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_delivery_d),
sizeof(m_txn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

		hr = pIAccessor->CreateAccessor(
			DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
			nOutputParams,
			acOutputDBBinding,
			sizeof(OL_ORDER_STATUS_DATA),
			&m_hOrderStatusOutputAccessor,
			acOutputDBBindStatus);
		if (FAILED(hr))
		{
			ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
		}

		// Now fill the binding information for result set 2 output
columns
		InitBindings(&acOutputDBBinding2[0], nOutputParams2,
eOutputColumn);

		i = 0;
		// OrderStatus output column 1
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_id), sizeof(m_txn.OrderStatus.c_id),
DBTYPE_I4);

		// OrderStatus output column 2
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

		// OrderStatus output column 3
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_txn.OrderStatus.c_first), DBTYPE_STR);

		// OrderStatus output column 4
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_txn.OrderStatus.c_middle), DBTYPE_STR);

		// OrderStatus output column 5
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_entry_d),
sizeof(m_txn.OrderStatus.o_entry_d), DBTYPE_DBTIMESTAMP);

		// OrderStatus output column 7
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_carrier_id),
sizeof(m_txn.OrderStatus.o_carrier_id), DBTYPE_I2);

		// OrderStatus output column 8
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_balance),
sizeof(m_txn.OrderStatus.c_balance), DBTYPE_R8);

		// OrderStatus output column 9
		SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_id), sizeof(m_txn.OrderStatus.o_id),
DBTYPE_I4);

		hr = pIAccessor->CreateAccessor(
			DBACCESSOR_ROWDATA, // cannot be optimized
too because #1 accessor is
			nOutputParams2,
			acOutputDBBinding2,
			sizeof(NEW_ORDER_DATA),
			&m_hOrderStatusOutputAccessor2,
			acOutputDBBindStatus2);
		if (FAILED(hr))
		{
			ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
		}
```

```
}

void CTPCC_OLEDB::OrderStatus()
{
	HRESULT                                         hr;
	int
	iTryCount = 0;
	IMultipleResults*	pMultipleResults;
	IRowset*                                pRowset;
	IRowset*                                pRowset2;
	LONG
	cRows = MAX_OL_ORDER_STATUS_ITEMS;	// number of rows returned in the 1st rowset
	ULONG
	cRowsObtained;
	HROW
	rghRows[MAX_OL_ORDER_STATUS_ITEMS];
	//returned row handles for the 1st result set
	HROW*
	prghRows = &rghRows[0];
	LONG
	cRows2 = 1; // number of rows returned in the 2nd rowset
	ULONG
	cRowsObtained2;
	HROW
	rghRows2;	//returned row handle for the 2nd result set
	HROW*
	prghRows2 = &rghRows2;
	int
	i;
	long
	lRowsAffected;	// the number of affected rows for a rowset

	if (m_txn.OrderStatus.c_id != 0)
		m_txn.OrderStatus.c_last[0] = 0;

	while (TRUE)
	{
		try
		{
			// Execute the prepared command
			// Ask for IMultipleResults because it returns 2 rowsets.
			hr = m_pIOrderStatusCommand->Execute(NULL, IID_IMultipleResults, &m_OrderStatusExecuteParams, NULL,
(IUnknown **)&pMultipleResults);
			if (FAILED(hr))
			{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eExecute, "OrderStatus()");
			}

			//////////////////////////
			// Get order line results
			//////////////////////////


			// Get the first rowset object
			hr = pMultipleResults->GetResult(NULL, 0, IID_IRowset, &lRowsAffected, (IUnknown **)&pRowset);
			if (FAILED(hr))
			{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetResult, "OrderStatus()");
			}

			// Fetch the result row handle(s)
			hr = pRowset->GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained, &prghRows);
			if (FAILED(hr))
			{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetNextRows, "OrderStatus()");
			}

			m_txn.OrderStatus.o_ol_cnt = (short)cRowsObtained;

			// Get the data from multiple rows in this rowset
			for (i = 0; i < m_txn.OrderStatus.o_ol_cnt; ++i)
			{
				// Fetch the actual row data by handle
				hr = pRowset->GetData(rghRows[i], m_hOrderStatusOutputAccessor, &m_txn.OrderStatus.OL[i]);
				if (FAILED(hr))
				{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetData, "OrderStatus()");
				}

				// Release row(s)
				hr = pRowset->ReleaseRows(cRowsObtained, prghRows, NULL, NULL, NULL);
				// Release rowset
				hr = pRowset->Release();

				////////////////////////////////
				// Get the second rowset object
				////////////////////////////////
				if (m_txn.OrderStatus.o_ol_cnt > 0)
				{
					hr = pMultipleResults->GetResult(NULL, 0, IID_IRowset, &lRowsAffected, (IUnknown **)&pRowset2);
					if (FAILED(hr))
					{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetResult, "OrderStatus()");
					}

					// Fetch the result row handle(s)
					hr = pRowset2->GetNextRows(DB_NULL_HCHAPTER, 0, cRows2, &cRowsObtained2, &prghRows2);
					if (FAILED(hr))
					{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetNextRows, "OrderStatus()");
					}

					// Fetch the actual row data by handle
					hr = pRowset2->GetData(rghRows2, m_hOrderStatusOutputAccessor2, &m_txn.OrderStatus);
					if (FAILED(hr))
					{

		ThrowError(m_pIOrderStatusCommand, COLEDBERR::eGetData, "OrderStatus()");
					}

					// Release row(s)
```

```
                                        hr = pRowset2->Release();
                                }

                                // Release the common MultipleResults interface
                                hr = pMultipleResults->Release();

                                if (m_txn.OrderStatus.o_ol_cnt == 0)
                                        throw new CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_NO_SUCH_ORDER );
                                else if (m_txn.OrderStatus.c_id == 0 && m_txn.OrderStatus.c_last[0] == 0)
                                        throw new CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
                                else

                m_txn.OrderStatus.exec_status_code = eOK;

                                break;

                        }
                        catch (COLEDBERR *e)
                        {
                                if ((!e->m_bDeadLock) || (++iTryCount > iMaxRetries))
                                        throw;

                                // hit deadlock; backoff for increasingly longer period
                                delete e;
                                Sleep(10 * iTryCount);
                        }
                }

//              if (iTryCount)
//                      throw new CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitDeliveryParams()
{
                int
                        i;
                HRESULT
                hr;
                wchar_t
                szName[iMAX_SP_NAME_LEN];
                IAccessor*
    const ULONG pIAccessor;
                nInputParams = 2;       // input parameters
                const ULONG
                nOutputParams = 10;     // output 1st result set columns

                // Structure to bind in accessor
                DBBINDING
                acInputDBBinding[nInputParams];
                DBBINDSTATUS
                acInputDBBindStatus[nInputParams];
                DBBINDING
                acOutputDBBinding[nOutputParams];
                DBBINDSTATUS
                acOutputDBBindStatus[nOutputParams];

                // Set command text
                _snwprintf(szName, sizeof(szName)/sizeof(szName[0]),
                                        L"{call %stpcc_delivery (?,?)}", m_szSPPrefix);

                // Create and Prepare a new command object for Delivery.

                CreateCommand(szName, &m_pIDeliveryCommand);

                // Describe the consumer buffer by filling in the array
    // of DBBINDING structures.  Each binding associates
    // a single parameter to the consumer's buffer.
                InitBindings(&acInputDBBinding[0], nInputParams, eInputParameter);

                i = 0;
    // Delivery parameter 1
                SetBinding(&acInputDBBinding[i++], offsetof(DELIVERY_DATA, w_id), sizeof(m_txn.Delivery.w_id), DBTYPE_I4);

    // Delivery parameter 2
                SetBinding(&acInputDBBinding[i++], offsetof(DELIVERY_DATA, o_carrier_id), sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);

                hr = m_pIDeliveryCommand->QueryInterface(IID_IAccessor, (void **)&pIAccessor);
                if (FAILED(hr))
                {
                        ThrowError(m_pIDeliveryCommand, COLEDBERR::eQueryInterface, "InitDeliveryParams()");
                }

                hr = pIAccessor->CreateAccessor(
                        DBACCESSOR_PARAMETERDATA,
                        nInputParams,
                        sizeof(DELIVERY_DATA),
                        &m_hDeliveryInputAccessor,
                        acInputDBBindStatus);
                if (FAILED(hr))
                {
                        ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
                }

                m_DeliveryExecuteParams.cParamSets = 1;
                m_DeliveryExecuteParams.hAccessor = m_hDeliveryInputAccessor;
                m_DeliveryExecuteParams.pData = &m_txn.Delivery;

                // Now fill the binding information for result set 1 output columns
                InitBindings(&acOutputDBBinding[0], nOutputParams, eOutputColumn);

                // Binding for a rowset that may return more than one row.
                for (i = 0; i < 10; ++i)
                {
                        // Delivery output column 1
                        SetBinding(&acOutputDBBinding[i], offsetof(DELIVERY_DATA, o_id[i]), sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
                }

                hr = pIAccessor->CreateAccessor(
                        DBACCESSOR_ROWDATA | DBACCESSOR_OPTIMIZED,
                        nOutputParams,
                        acOutputDBBinding,
                        sizeof(DELIVERY_DATA),
                        &m_hDeliveryOutputAccessor,
                        acOutputDBBindStatus);
                if (FAILED(hr))
                {
                        ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
                }
}

void CTPCC_OLEDB::Delivery()
{
                HRESULT                         hr;
                int
                iTryCount = 0;
                IRowset*                pRowset;
                LONG                            cRows = 1;   // number of rows returned in the rowset
                ULONG                           cRowsObtained;
```

```
		HROW				rghRow;
		//returned row handles
		HROW*				prghRow =
&rghRow;

		while (TRUE)
		{
			try
			{
				// Execute the prepared
command
				hr = m_pIDeliveryCommand-
>Execute(NULL, IID_IRowset, &m_DeliveryExecuteParams, NULL,


		(IUnknown **)&pRowset);
				if (FAILED(hr))
				{

		ThrowError(m_pIDeliveryCommand,
COLEDBERR::eExecute, "Delivery()");
				}

				// Fetch the result row
handle(s)
				hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows, &cRowsObtained,
&prghRow);
				if (FAILED(hr))
				{

		ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetNextRows, "Delivery()");
				}

				// Fetch the actual row data by
handle
				hr = pRowset-
>GetData(rghRow, m_hDeliveryOutputAccessor, &m_txn.Delivery);
				if (FAILED(hr))
				{

		ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetData, "Delivery()");
				}

				// Release row(s)
				hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL, NULL);
				// Release rowset
				hr = pRowset->Release();

		m_txn.Delivery.exec_status_code = eOK;
				break;
			}
			catch (COLEDBERR *e)
			{
				if ((!e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
					throw;

				// hit deadlock; backoff for
increasingly longer period
				delete e;
				Sleep(10 * iTryCount);
			}
		}

//		if (iTryCount)
//			throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}
```

tm_com_dll\src\tpcc_com.cpp

```
/*		FILE:			TPCC_COM.CPP
 *				Microsoft TPC-C
Kit Ver. 4.20.000
 *				Copyright
Microsoft, 1999
 *		All Rights Reserved
 *
 *				not yet audited
 *
 *		PURPOSE:	Source file for TPC-C COM+ class
implementation.
 *		Contact:	Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *			4.20.000 - first version
 */

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

// need to declare functions for export
#define DllDecl  __declspec( dllexport )

#include "..\..\common\src\trans.h"			//tpckit
transaction header contains definations of structures specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_com.h"

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL
bSinglePool)
{
		return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
		HRESULT hr = NULL;
		long lRet = 0;
		ULONG ulTmpSize = 0;

		m_pTxn				= NULL;
		m_pNewOrder			= NULL;
		m_pPayment		= NULL;
		m_pStockLevel		= NULL;
		m_pOrderStatus		= NULL;

		m_bSinglePool		= bSinglePool;

		ulTmpSize = (ULONG) sizeof(COM_DATA);
		VariantInit(&m_vTxn);
		m_vTxn.vt = VT_SAFEARRAY;

		m_vTxn.parray = SafeArrayCreateVector(VT_UI1,
ulTmpSize, ulTmpSize);
		if (!m_vTxn.parray)
			throw new CCOMERR( E_FAIL );

		memset((void*)m_vTxn.parray->pvData,0,ulTmpSize);
		m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

		hr = CoInitializeEx(NULL, COINIT_MULTITHREADED);
		if (FAILED(hr))
		{
			throw new CCOMERR( hr );
		}

		// create components
		if (m_bSinglePool)
		{
			hr = CoCreateInstance(CLSID_TPCC, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
```

```
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        // all txns will use same component
                        m_pPayment = m_pNewOrder;
                        m_pStockLevel = m_pNewOrder;
                        m_pOrderStatus = m_pNewOrder;
        }
        else
        {
                        // use different components for each txn

                        hr = CoCreateInstance(CLSID_NewOrder,
NULL, CLSCTX_SERVER, IID_ITPCC, (void **)&m_pNewOrder);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_Payment,
NULL, CLSCTX_SERVER, IID_ITPCC, (void **)&m_pPayment);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_StockLevel,
NULL, CLSCTX_SERVER, IID_ITPCC, (void **)&m_pStockLevel);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = CoCreateInstance(CLSID_OrderStatus,
NULL, CLSCTX_SERVER, IID_ITPCC, (void **)&m_pOrderStatus);
                        if (FAILED(hr))
                                throw new CCOMERR(hr);
        }

        // call setcomplete to release each component back into
pool
        hr = m_pNewOrder->CallSetComplete();
        if (FAILED(hr))
                        throw new CCOMERR(hr);

        if (!m_bSinglePool)
        {
                        hr = m_pPayment->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = m_pStockLevel->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);

                        hr = m_pOrderStatus->CallSetComplete();
                        if (FAILED(hr))
                                throw new CCOMERR(hr);
        }
}

CTPCC_COM::~CTPCC_COM()
{
        if (m_pTxn)
                SafeArrayDestroy(m_vTxn.parray);

        ReleaseInterface(m_pNewOrder);
        if (!m_bSinglePool)
        {
                ReleaseInterface(m_pPayment);
                ReleaseInterface(m_pStockLevel);
                ReleaseInterface(m_pOrderStatus);
        }
        CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
        VARIANT   vTxn_out;

        HRESULT hr = m_pNewOrder->NewOrder(m_vTxn,
&vTxn_out);
        if (FAILED(hr))
                throw new CCOMERR( hr );
        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        SafeArrayDestroy(vTxn_out.parray);

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
                throw new CCOMERR( m_pTxn->ErrorType,
m_pTxn->error );
}

void CTPCC_COM::Payment()
{
        VARIANT   vTxn_out;

        HRESULT hr = m_pPayment->Payment(m_vTxn,
&vTxn_out);
        if (FAILED(hr))
                throw new CCOMERR( hr );
        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        SafeArrayDestroy(vTxn_out.parray);

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
                throw new CCOMERR( m_pTxn->ErrorType,
m_pTxn->error );
}

void CTPCC_COM::StockLevel()
{
        VARIANT   vTxn_out;
        HRESULT hr = m_pStockLevel->StockLevel(m_vTxn,
&vTxn_out);
        if (FAILED(hr))
                throw new CCOMERR( hr );
        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        SafeArrayDestroy(vTxn_out.parray);

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
                throw new CCOMERR( m_pTxn->ErrorType,
m_pTxn->error );
}

void CTPCC_COM::OrderStatus()
{
        VARIANT   vTxn_out;

        HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn,
&vTxn_out);
        if (FAILED(hr))
                throw new CCOMERR( hr );
        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        SafeArrayDestroy(vTxn_out.parray);

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
                throw new CCOMERR( m_pTxn->ErrorType,
m_pTxn->error );
}
```

\tpcc_com_all\src\methods.h

```
/*      FILE:                   METHODS.H
 *                                              Microsoft TPC-C
Kit Ver. 4.20.000
 *                                              Copyright
Microsoft, 1999
 *      All Rights Reserved
 *
 *                                              not yet audited
 *
 *      PURPOSE:    Header file for COM components.
 *
 *  Change history:
 *                  4.20.000 - first version
 */
```

```
enum COMPONENT_ERROR
{
	ERR_MISSING_REGISTRY_ENTRIES = 1,
	ERR_LOADDLL_FAILED,
	ERR_GETPROCADDR_FAILED,
	ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
	public:
		CCOMPONENT_ERR(COMPONENT_ERROR
Err)
		{
			m_Error = Err;
			m_szTextDetail = NULL;
			m_SystemErr = 0;
			m_szErrorText = NULL;
		};

		CCOMPONENT_ERR(COMPONENT_ERROR
Err, char *szTextDetail, DWORD dwSystemErr)
		{
			m_Error = Err;
			m_szTextDetail = new
char[strlen(szTextDetail)+1];
			strcpy( m_szTextDetail,
szTextDetail );
			m_SystemErr = dwSystemErr;
			m_szErrorText = NULL;
		};

		~CCOMPONENT_ERR()
		{
			if (m_szTextDetail != NULL)
				delete []
m_szTextDetail;
			if (m_szErrorText != NULL)
				delete []
m_szErrorText;
		};

		COMPONENT_ERROR	m_Error;
		char
	*m_szTextDetail;
		char
	*m_szErrorText;
		DWORD
	m_SystemErr;
		int ErrorType() {return
ERR_TYPE_COMPONENT;};char *ErrorTypeStr() { return
"COMPONENT"; }
		int ErrorNum() {return m_Error;};
		char *ErrorText();
};


static void WriteMessageToEventLog(LPTSTR lpszMsg);


/////////////////////////////////////////////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
	public ITPCC,
	public IObjectControl,
	public IObjectConstruct,
	public CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
	COM_INTERFACE_ENTRY(ITPCC)
	COM_INTERFACE_ENTRY(IObjectControl)
	COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

	CTPCC_Common();
	~CTPCC_Common();

// ITPCC
public:
	HRESULT __stdcall NewOrder(
	VARIANT txn_in, VARIANT* txn_out);
	HRESULT __stdcall Payment(
	VARIANT txn_in, VARIANT* txn_out);
	HRESULT __stdcall Delivery(
	VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
	HRESULT __stdcall StockLevel(	VARIANT txn_in,
VARIANT* txn_out);
	HRESULT __stdcall OrderStatus(	VARIANT txn_in,
VARIANT* txn_out);

	HRESULT __stdcall CallSetComplete();

// IObjectControl
	STDMETHODIMP_(BOOL) CanBePooled() { return
m_bCanBePooled; }
	STDMETHODIMP Activate() { return S_OK; }	//
we don't support COM Services transactions (no enlistment)
	STDMETHODIMP_(void) Deactivate() { /* nothing to do */
}
// IObjectConstruct
	STDMETHODIMP Construct(IDispatch * pUnk);

	// helper methods
private:
	BOOL			m_bCanBePooled;
	CTPCC_BASE *m_pTxn;

	struct COM_DATA
	{
		int retval;
		int error;
		union
		{
			NEW_ORDER_DATA
	NewOrder;
			PAYMENT_DATA
	Payment;
			DELIVERY_DATA
	Delivery;
			STOCK_LEVEL_DATA
	StockLevel;
			ORDER_STATUS_DATA
	OrderStatus;
		} u;
	};

};

/////////////////////////////////////////////////////////////////////////////
// CTPCC
class CTPCC :
	public CTPCC_Common,
	public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

BEGIN_COM_MAP(CTPCC)
	//COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx<CComSingleThreadModel>)
	COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
	COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


/////////////////////////////////////////////////////////////////////////////
// CNewOrder
class CNewOrder :
	public CTPCC_Common,
	public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
```

```
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

BEGIN_COM_MAP(CNewOrder)
//              COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
                COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
                COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
//              HRESULT __stdcall NewOrder(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall Payment(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall StockLevel(           VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
                HRESULT __stdcall OrderStatus(          VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
};


                ////////////////////////////////////////////////////////////////
/////////
// COrderStatus
class COrderStatus :
                public CTPCC_Common,
                public CComCoClass<COrderStatus, &CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

BEGIN_COM_MAP(COrderStatus)
//              COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
                COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
                COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
                HRESULT __stdcall NewOrder(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall Payment(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall StockLevel(           VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
//              HRESULT __stdcall OrderStatus(          VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
};


////////////////////////////////////////////////////////////////////////
// CPayment
class CPayment :
                public CTPCC_Common,
                public CComCoClass<CPayment, &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
//              COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
                COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
                COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
                HRESULT __stdcall NewOrder(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
//              HRESULT __stdcall Payment(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall StockLevel(           VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
                HRESULT __stdcall OrderStatus(          VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
};


////////////////////////////////////////////////////////////////////////
// CStockLevel
class CStockLevel :
                public CTPCC_Common,
                public CComCoClass<CStockLevel, &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
//              COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
                COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
                COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
                HRESULT __stdcall NewOrder(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
                HRESULT __stdcall Payment(
                VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
//              HRESULT __stdcall StockLevel(           VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
                HRESULT __stdcall OrderStatus(          VARIANT txn_in,
VARIANT* txn_out) {return E_NOTIMPL;}
};
```

\tpcc_com_all\src\resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME                100
#define IDR_TPCC                    101
#define IDR_NEWORDER                102
#define IDR_ORDERSTATUS             103
#define IDR_PAYMENT                 104
#define IDR_STOCKLEVEL              105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE    202
#define _APS_NEXT_COMMAND_VALUE     32768
#define _APS_NEXT_CONTROL_VALUE     201
#define _APS_NEXT_SYMED_VALUE       106
#endif
#endif
```

\tpcc_com_all\src\tpcc_com_all.cpp

```
/*              FILE:                   TPCC_COM_ALL.CPP
 *                                                      Microsoft TPC-C
Kit Ver. 4.20.000
```

```
 *                                                      Copyright
Microsoft, 1999
 *           All Rights Reserved
 *
 *                                                      Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *           PURPOSE:    Implementation for TPC-C Tuxedo class.
 *           Contact:    Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                       4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to
override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
                        //tpckit transaction header contains
definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  //
DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    //
ODBC implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"

#include "..\..\common\src\ReadRegistry.cpp"

CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
        OBJECT_ENTRY(CLSID_TPCC, CTPCC)
        OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
        OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
        OBJECT_ENTRY(CLSID_Payment, CPayment)
        OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()


// configuration settings from registry
TPCCREGISTRYDATA        Reg;
char
        szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB        *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;

// Critical section to synchronize connection open and close.
//
CRITICAL_SECTION hConnectCriticalSection;

////////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason,
LPVOID /*lpReserved*/)
{
        char szDllName[128];

        try
        {
                if (dwReason == DLL_PROCESS_ATTACH)
                {
                        _Module.Init(ObjectMap,
hInstance);

                DisableThreadLibraryCalls(hInstance);

                        DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;

                GetComputerName(szMyComputerName, &dwSize);
                        szMyComputerName[dwSize] =
0;

                        if ( ReadTPCCRegistrySettings(
&Reg ) )
                                throw new
CCOMPONENT_ERR( ERR_MISSING_REGISTRY_ENTRIES );

                        if (Reg.eDB_Protocol ==
DBLIB)
                        {
                                strcpy(
szDllName, Reg.szPath );
                                strcat(
szDllName, "tpcc_dblib.dll");
                                hLibInstanceDb =
LoadLibrary( szDllName );
                                if
(hLibInstanceDb == NULL)

                throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                // get function
pointer to wrapper for class constructor

                pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                if
(pCTPCC_DBLIB_new == NULL)

                throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                        }
                        else if (Reg.eDB_Protocol ==
ODBC)
                        {
                                strcpy(
szDllName, Reg.szPath );
                                strcat(
szDllName, "tpcc_odbc.dll");
                                hLibInstanceDb =
LoadLibrary( szDllName );
                                if
(hLibInstanceDb == NULL)

                throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                // get function
pointer to wrapper for class constructor

                pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                if
(pCTPCC_ODBC_new == NULL)
```

```
            throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                    }
                                    else
                                                throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );

                                    if (Reg.dwConnectDelay > 0)
                                    {

            InitializeCriticalSection(&hConnectCriticalSection);
                                    }

                        }
                        else if (dwReason ==
DLL_PROCESS_DETACH)
                                    _Module.Term();

            }
            catch (CBaseErr *e)
            {
                        TCHAR szMsg[256];

                        _sntprintf(szMsg, sizeof(szMsg), "%s error,
code %d: %s",
                                                e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                        WriteMessageToEventLog( szMsg );

                        delete e;
                        return FALSE;
            }
            catch (...)
            {
                        WriteMessageToEventLog(TEXT("Unhandled
exception in object DllMain"));
                        return FALSE;
            }

            return TRUE; // OK
}

////////////////////////////////////////////////////////////////////////////
// Used to determine whether the DLL can be unloaded by OLE

STDAPI DllCanUnloadNow(void)
{
            return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
}

////////////////////////////////////////////////////////////////////////////
```

```
// Returns a class factory to create an object of the requested type
STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
            return _Module.GetClassObject(rclsid, riid, ppv);
}

////////////////////////////////////////////////////////////////////////////
// DllRegisterServer - Adds entries to the system registry

STDAPI DllRegisterServer(void)
{
            // registers object, typelib and all interfaces in typelib
            return _Module.RegisterServer(TRUE);
}

////////////////////////////////////////////////////////////////////////////
// DllUnregisterServer - Removes entries from the system registry

STDAPI DllUnregisterServer(void)
{
            _Module.UnregisterServer();
            return S_OK;
}


static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL,
TEXT("tpcc_com_all.dll"));

   _stprintf(szMsg, TEXT("Error in COM+ TPC-C Component: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
            {
      ReportEvent(hEventSource, // handle of event source
         EVENTLOG_ERROR_TYPE,  // event type
         0,                    // event category
         0,                    // event ID
         NULL,                 // current user's SID
         2,                    // strings in lpszStrings
         0,                    // no bytes of raw data
         (LPCTSTR *)lpszStrings,   // array of error strings
         NULL);                // no raw data

      (VOID) DeregisterEventSource(hEventSource);
```

```
} }

inline void ReleaseInterface(IUnknown *pUnk)
{
            if (pUnk)
            {
                        pUnk->Release();
                        pUnk = NULL;
            }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

char* CCOMPONENT_ERR::ErrorText(void)
{
            static SERRORMSG errorMsgs[] =
            {
                        { ERR_MISSING_REGISTRY_ENTRIES,
            "Required entries missing from registry."
                        },
                        { ERR_LOADDLL_FAILED,
            "Load of DLL failed.  DLL="

            },
                        { ERR_GETPROCADDR_FAILED,
            "Could not map proc in DLL.  GetProcAddr error.  DLL="
            },
                        { ERR_UNKNOWN_DB_PROTOCOL,
            "Unknown database protocol specified in registry."
            },
                        { 0,
                                                ""


                                    }
            };

            char szTmp[256];
            int i = 0;
            while (TRUE)
            {
                        if (errorMsgs[i].szMsg[0] == 0)
                        {
                                    strcpy( szTmp, "Unknown error
number." );
                                    break;
                        }
                        if (m_Error == errorMsgs[i].iError)
                        {
                                    strcpy( szTmp,
errorMsgs[i].szMsg );
```

```
                                        break;
                                }
                                i++;
                        }

                        if (m_szTextDetail)
                                strcat( szTmp, m_szTextDetail );
                        if (m_SystemErr)
                                wsprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

                        m_szErrorText = new char[strlen(szTmp)+1];
                        strcpy( m_szErrorText, szTmp );
                        return m_szErrorText;
}


CTPCC_Common::CTPCC_Common()
{
                        m_pTxn = NULL;
                        m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
                        // Pace connection close for VIA.
                        //
                        if (Reg.dwConnectDelay > 0)
                        {

                        EnterCriticalSection(&hConnectCriticalSection);

                                Sleep(Reg.dwConnectDelay);
                        }

                        if (m_pTxn)
                        {
                                delete m_pTxn;
                        }

                        if (Reg.dwConnectDelay > 0)
                        {

                        LeaveCriticalSection(&hConnectCriticalSection);
                        }
}

HRESULT CTPCC_Common::CallSetComplete()
{
                        IObjectContext* pObjectContext = NULL;

                        // get our object context

                        HRESULT hr = CoGetObjectContext( IID_IObjectContext,
(void **)&pObjectContext);
                        pObjectContext->SetComplete();
                        ReleaseInterface(pObjectContext);
                        return hr;
}

//
//              called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
                        // Code to access construction string, if needed later...
                        //              if (!pUnk)
                        //                      return E_UNEXPECTED;
                        //              IObjectConstructString * pString = NULL;
                        //              HRESULT hr = pUnk-
>QueryInterface(IID_IObjectConstructString, (void **)&pString);
                        //              pString->Release();

                        try
                        {
                                // Pace connection creation for VIA.
                                //
                                if (Reg.dwConnectDelay > 0)
                                {

                        EnterCriticalSection(&hConnectCriticalSection);

                                        Sleep(Reg.dwConnectDelay);
                                }

                                if (Reg.eDB_Protocol == ODBC)
                                        m_pTxn =
pCTPCC_ODBC_new(        Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,


                        szMyComputerName, Reg.szDbName,


                        Reg.szSPPrefix, Reg.bCallNoDuplicatesNewOrder );
                                else if (Reg.eDB_Protocol == DBLIB)
                                        m_pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );

                                if (Reg.dwConnectDelay > 0)
                                {

                        LeaveCriticalSection(&hConnectCriticalSection);
                                }
                        }
                        catch (CBaseErr *e)
                        {               TCHAR szMsg[256];

                                _sntprintf(szMsg, sizeof(szMsg), "%s error in
CTPCC_Common::Construct, code %d: %s",
                                                                e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                                WriteMessageToEventLog( szMsg );
                                delete e;
                                return E_FAIL;
                        }
                        catch (...)
                        {
                                WriteMessageToEventLog(TEXT("Unhandled
exception in object ::Construct"));
                                return E_FAIL;
                        }

                        return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in, VARIANT*
txn_out)
{
                        PNEW_ORDER_DATA         pNewOrder;
                        COM_DATA                *pData;
                        try
                        {
                                pData = (COM_DATA*)txn_in.parray-
>pvData;
                                pNewOrder = m_pTxn-
>BuffAddr_NewOrder();

                                memcpy(pNewOrder, &pData->u.NewOrder,
sizeof(NEW_ORDER_DATA));

                                m_pTxn->NewOrder();                     //
do the actual txn

                                VariantInit(txn_out);
                                txn_out->vt = VT_SAFEARRAY;
                                txn_out->parray =
SafeArrayCreateVector(VT_UI1,

                                txn_in.parray->rgsabound->cElements,

                                txn_in.parray->rgsabound->cElements);
                                pData = (COM_DATA*) txn_out->parray-
>pvData;

                                memcpy( &pData->u.NewOrder,
pNewOrder, sizeof(NEW_ORDER_DATA));

                                pData->retval = ERR_SUCCESS;
```

```
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes,
component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB)
&& (e->ErrorNum() == 10005)) ||
                                ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled
exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in, VARIANT*
txn_out)
{
        PPAYMENT_DATA           pPayment;
        COM_DATA                *pData;
        try
        {
                pData = (COM_DATA*)txn_in.parray-
>pvData;
                pPayment = m_pTxn->BuffAddr_Payment();

                memcpy(pPayment, &pData->u.Payment,
sizeof(PAYMENT_DATA));

                m_pTxn->Payment();              //
do the actual txn

                VariantInit(txn_out);
                txn_out->vt = VT_SAFEARRAY;
                txn_out->parray = SafeArrayCreateVector(
VT_UI1,

                txn_in.parray->rgsabound->cElements,
                pData = (COM_DATA*)txn_in.parray-
>pvData;

                memcpy( &pData->u.Payment, pPayment,
sizeof(PAYMENT_DATA));

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes,
component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB)
&& (e->ErrorNum() == 10005)) ||
                                ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled
exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::StockLevel(VARIANT txn_in, VARIANT*
txn_out)
{
        PSTOCK_LEVEL_DATA       pStockLevel;
        COM_DATA                        *pData;

        try
        {
                pData = (COM_DATA*)txn_in.parray-
>pvData;
                pStockLevel = m_pTxn-
>BuffAddr_StockLevel();

                memcpy(pStockLevel, &pData-
>u.StockLevel, sizeof(STOCK_LEVEL_DATA));
                m_pTxn->StockLevel();
                VariantInit(txn_out);
                txn_out->vt = VT_SAFEARRAY;
                txn_out->parray = SafeArrayCreateVector(
VT_UI1,

                txn_in.parray->rgsabound->cElements,

                txn_in.parray->rgsabound->cElements);
                pData = (COM_DATA*)txn_out->parray-
>pvData;

                memcpy( &pData->u.StockLevel,
pStockLevel, sizeof(STOCK_LEVEL_DATA));

                pData->retval = ERR_SUCCESS;
                pData->error = 0;
                return S_OK;
        }
        catch (CBaseErr *e)
        {
                // check for lost database connection; if yes,
component is toast
                if ( ((e->ErrorType() == ERR_TYPE_DBLIB)
&& (e->ErrorNum() == 10005)) ||
                                ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                pData->retval = e->ErrorType();
                pData->error = e->ErrorNum();
                delete e;
                return E_FAIL;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled
exception."));
                pData->retval = ERR_TYPE_LOGIC;
                pData->error = 0;
                m_bCanBePooled = FALSE;
                return E_FAIL;
        }
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in, VARIANT*
txn_out)
{
        PORDER_STATUS_DATA      pOrderStatus;
        COM_DATA                        *pData;
        try
        {
```

```
                        pData = (COM_DATA*)txn_in.parray->pvData;
                        pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

                        memcpy(pOrderStatus, &pData->u.OrderStatus, sizeof(ORDER_STATUS_DATA));

                        m_pTxn->OrderStatus();

                        VariantInit(txn_out);
                        txn_out->vt = VT_SAFEARRAY;
                        txn_out->parray = SafeArrayCreateVector( VT_UI1,

                        txn_in.parray->rgsabound->cElements,

                        txn_in.parray->rgsabound->cElements);
                        pData = (COM_DATA*)txn_out->parray->pvData;

                        memcpy( &pData->u.OrderStatus, pOrderStatus, sizeof(ORDER_STATUS_DATA));

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
                }
                catch (CBaseErr *e)
                {
                        // check for lost database connection; if yes, component is toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
                }
}
```

\tpcc_com_all\src\tpcc_com_all.def

```
/*              FILE:                   TPCC_COM_ALL.CPP
 *                                              Microsoft TPC-C Kit Ver. 4.20.000
 *                                              Copyright Microsoft, 1999
 *              All Rights Reserved
 *
 *                                              Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *              PURPOSE:        Implementation for TPC-C Tuxedo class.
 *              Contact:        Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                      4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
                        //tpckit transaction header contains definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"

#include "..\..\db_dblib_dll\src\tpcc_dblib.h"                  // DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                    // ODBC implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
                OBJECT_ENTRY(CLSID_TPCC, CTPCC)
                OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
                OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
                OBJECT_ENTRY(CLSID_Payment, CPayment)
                OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()


// configuration settings from registry
TPCCREGISTRYDATA        Reg;
char
                szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB        *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC                 *pCTPCC_ODBC_new;

// Critical section to synchronize connection open and close.
//
CRITICAL_SECTION hConnectCriticalSection;

////////////////////////////////////////////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason,
LPVOID /*lpReserved*/)
{
                char szDllName[128];

                try
                {
                        if (dwReason == DLL_PROCESS_ATTACH)
                        {
                                _Module.Init(ObjectMap, hInstance);
```

```
		DisableThreadLibraryCalls(hInstance);

                                DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;

		GetComputerName(szMyComputerName, &dwSize);
                                szMyComputerName[dwSize] =
0;

                                if ( ReadTPCCRegistrySettings(
&Reg ) )
                                        throw new
CCOMPONENT_ERR( ERR_MISSING_REGISTRY_ENTRIES );

                                if (Reg.eDB_Protocol ==
DBLIB)
                                {
                                        strcpy(
szDllName, Reg.szPath );
                                        strcat(
szDllName, "tpcc_dblib.dll");
                                        hLibInstanceDb =
LoadLibrary( szDllName );
                                        if
(hLibInstanceDb == NULL)

		throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                        // get function
pointer to wrapper for class constructor

		pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                        if
(pCTPCC_DBLIB_new == NULL)

		throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                }
                                else if (Reg.eDB_Protocol ==
ODBC)
                                {
                                        strcpy(
szDllName, Reg.szPath );
                                        strcat(
szDllName, "tpcc_odbc.dll");
                                        hLibInstanceDb =
LoadLibrary( szDllName );
                                        if
(hLibInstanceDb == NULL)

		throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );
                                        // get function
pointer to wrapper for class constructor

		pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                        if
(pCTPCC_ODBC_new == NULL)

		throw new CCOMPONENT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
                                }
                                else
                                        throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );

                                if (Reg.dwConnectDelay > 0)
                                {

		InitializeCriticalSection(&hConnectCriticalSection);
                                }

                        }
                        else if (dwReason ==
DLL_PROCESS_DETACH)
                                _Module.Term();

                }
                catch (CBaseErr *e)
                {
                        TCHAR szMsg[256];

                        _sntprintf(szMsg, sizeof(szMsg), "%s error,
code %d: %s",
                                        e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                        WriteMessageToEventLog( szMsg );

                        delete e;
                        return FALSE;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled
exception in object DllMain"));
                        return FALSE;
                }

                return TRUE; // OK
}
```

```
// Used to determine whether the DLL can be unloaded by OLE
/////////////////////////////////////////////////////////////////

STDAPI DllCanUnloadNow(void)
{
        return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
}

/////////////////////////////////////////////////////////////////////////
// Returns a class factory to create an object of the requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID riid, LPVOID* ppv)
{
        return _Module.GetClassObject(rclsid, riid, ppv);
}

/////////////////////////////////////////////////////////////////////////
// DllRegisterServer - Adds entries to the system registry

STDAPI DllRegisterServer(void)
{
        // registers object, typelib and all interfaces in typelib
        return _Module.RegisterServer(TRUE);
}

/////////////////////////////////////////////////////////////////////////
// DllUnregisterServer - Removes entries from the system registry

STDAPI DllUnregisterServer(void)
{
        _Module.UnregisterServer();
        return S_OK;
}


static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
   TCHAR   szMsg[256];
   HANDLE  hEventSource;
   LPTSTR  lpszStrings[2];

   // Use event logging to log the error.
   //
   hEventSource = RegisterEventSource(NULL,
TEXT("tpcc_com_all.dll"));

   _stprintf(szMsg, TEXT("Error in COM+ TPC-C Component: "));
   lpszStrings[0] = szMsg;
   lpszStrings[1] = lpszMsg;

   if (hEventSource != NULL)
        {
     ReportEvent(hEventSource, // handle of event source
        EVENTLOG_ERROR_TYPE,  // event type
```

```
        0,              // event category
        0,              // event ID
        NULL,           // current user's SID
        2,              // strings in lpszStrings
        0,              // no bytes of raw data
        (LPCTSTR *)lpszStrings,   // array of error strings
        NULL);          // no raw data

    (VOID) DeregisterEventSource(hEventSource);
  }
}

inline void ReleaseInterface(IUnknown *pUnk)
{
        if (pUnk)
        {
                pUnk->Release();
                pUnk = NULL;
        }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

char* CCOMPONENT_ERR::ErrorText(void)
{
        static SERRORMSG errorMsgs[] =
        {
                { ERR_MISSING_REGISTRY_ENTRIES,
        "Required entries missing from registry."
                },
                { ERR_LOADDLL_FAILED,
        "Load of DLL failed.  DLL="

        },
                { ERR_GETPROCADDR_FAILED,
        "Could not map proc in DLL.  GetProcAddr error.  DLL="
        },
                { ERR_UNKNOWN_DB_PROTOCOL,
        "Unknown database protocol specified in registry."
        },
                { 0,
                                        ""


                                }
        };

        char szTmp[256];
        int i = 0;
        while (TRUE)
                {       if (errorMsgs[i].szMsg[0] == 0)
                        {
                                strcpy( szTmp, "Unknown error
number." );
                                break;
                        }
                        if (m_Error == errorMsgs[i].iError)
                        {
                                strcpy( szTmp,
errorMsgs[i].szMsg );
                                break;
                        }
                        i++;
                }

                if (m_szTextDetail)
                        strcat( szTmp, m_szTextDetail );
                if (m_SystemErr)
                        wsprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

                m_szErrorText = new char[strlen(szTmp)+1];
                strcpy( m_szErrorText, szTmp );
                return m_szErrorText;
}


CTPCC_Common::CTPCC_Common()
{
        m_pTxn = NULL;
        m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
        // Pace connection close for VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {

        EnterCriticalSection(&hConnectCriticalSection);

                Sleep(Reg.dwConnectDelay);
        }

        if (m_pTxn)
        {
                delete m_pTxn;
        }

        if (Reg.dwConnectDelay > 0)
        {
                LeaveCriticalSection(&hConnectCriticalSection);
}

HRESULT CTPCC_Common::CallSetComplete()
{
        IObjectContext* pObjectContext = NULL;

        // get our object context
        HRESULT hr = CoGetObjectContext( IID_IObjectContext,
(void **)&pObjectContext );
        pObjectContext->SetComplete();
        ReleaseInterface(pObjectContext);
        return hr;
}

//
//      called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
        // Code to access construction string, if needed later...
        //      if (!pUnk)
        //              return E_UNEXPECTED;
        //      IObjectConstructString * pString = NULL;
        //      HRESULT hr = pUnk-
>QueryInterface(IID_IObjectConstructString, (void **)&pString);
        //      pString->Release();

        try
        {
                // Pace connection creation for VIA.
                //
                if (Reg.dwConnectDelay > 0)
                {

        EnterCriticalSection(&hConnectCriticalSection);

                        Sleep(Reg.dwConnectDelay);
                }

                if (Reg.eDB_Protocol == ODBC)
                        m_pTxn =
pCTPCC_ODBC_new(        Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,


        szMyComputerName, Reg.szDbName,


        Reg.szSPPrefix, Reg.bCallNoDuplicatesNewOrder );
                else if (Reg.eDB_Protocol == DBLIB)
```

```
                                        m_pTxn =
pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );

                        if (Reg.dwConnectDelay > 0)
                        {

        LeaveCriticalSection(&hConnectCriticalSection);
                        }
        }
        catch (CBaseErr *e)
        {
                        TCHAR szMsg[256];

                        _sntprintf(szMsg, sizeof(szMsg), "%s error in
CTPCC_Common::Construct, code %d: %s",
                                                e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
                        WriteMessageToEventLog( szMsg );
                        delete e;
                        return E_FAIL;
        }
        catch (...)
        {
                        WriteMessageToEventLog(TEXT("Unhandled
exception in object ::Construct"));
                        return E_FAIL;
        }

        return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in, VARIANT*
txn_out)
{
        PNEW_ORDER_DATA         pNewOrder;
        COM_DATA                *pData;
        try
        {
                        pData = (COM_DATA*)txn_in.parray-
>pvData;
                        pNewOrder = m_pTxn-
>BuffAddr_NewOrder();

                        memcpy(pNewOrder, &pData->u.NewOrder,
sizeof(NEW_ORDER_DATA));

                        m_pTxn->NewOrder();             //
do the actual txn

                        VariantInit(txn_out);
                        txn_out->vt = VT_SAFEARRAY;
                        txn_out->parray =
SafeArrayCreateVector(VT_UI1,
                        txn_in.parray->rgsabound->cElements,

                        txn_in.parray->rgsabound->cElements);
                        pData = (COM_DATA*) txn_out->parray-
>pvData;

                        memcpy( &pData->u.NewOrder,
pNewOrder, sizeof(NEW_ORDER_DATA));

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
        }
        catch (CBaseErr *e)
        {
                        // check for lost database connection; if yes,
component is toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB)
&& (e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                        m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
        }
        catch (...)
        {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
        }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in, VARIANT*
txn_out)
{
        PPAYMENT_DATA           pPayment;
        COM_DATA                *pData;
        try
        {
                        pData = (COM_DATA*)txn_in.parray-
>pvData;
                        pPayment = m_pTxn->BuffAddr_Payment();
                        memcpy(pPayment, &pData->u.Payment,
sizeof(PAYMENT_DATA));
                        m_pTxn->Payment();              //
do the actual txn

                        VariantInit(txn_out);
                        txn_out->vt = VT_SAFEARRAY;
                        txn_out->parray = SafeArrayCreateVector(
VT_UI1,

                        txn_in.parray->rgsabound->cElements,

                        txn_in.parray->rgsabound->cElements);
                        pData = (COM_DATA*) txn_out->parray-
>pvData;

                        memcpy( &pData->u.Payment, pPayment,
sizeof(PAYMENT_DATA));

                        pData->retval = ERR_SUCCESS;
                        pData->error = 0;
                        return S_OK;
        }
        catch (CBaseErr *e)
        {
                        // check for lost database connection; if yes,
component is toast
                        if ( ((e->ErrorType() == ERR_TYPE_DBLIB)
&& (e->ErrorNum() == 10005)) ||
                                        ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                                        m_bCanBePooled = FALSE;

                        pData->retval = e->ErrorType();
                        pData->error = e->ErrorNum();
                        delete e;
                        return E_FAIL;
        }
        catch (...)
        {
                        WriteMessageToEventLog(TEXT("Unhandled
exception."));
                        pData->retval = ERR_TYPE_LOGIC;
                        pData->error = 0;
                        m_bCanBePooled = FALSE;
                        return E_FAIL;
        }
}

HRESULT CTPCC_Common::StockLevel(VARIANT txn_in, VARIANT*
txn_out)
{
        PSTOCK_LEVEL_DATA       pStockLevel;
```

```
		COM_DATA				*pData;

		try
		{
			pData = (COM_DATA*)txn_in.parray->pvData;
			pStockLevel = m_pTxn->BuffAddr_StockLevel();

			memcpy(pStockLevel, &pData->u.StockLevel, sizeof(STOCK_LEVEL_DATA));

			m_pTxn->StockLevel();

			VariantInit(txn_out);
			txn_out->vt = VT_SAFEARRAY;
			txn_out->parray = SafeArrayCreateVector( VT_UI1,

			txn_in.parray->rgsabound->cElements,

			txn_in.parray->rgsabound->cElements);
			pData = (COM_DATA*)txn_out->parray->pvData;

			memcpy( &pData->u.StockLevel, pStockLevel, sizeof(STOCK_LEVEL_DATA));

			pData->retval = ERR_SUCCESS;
			pData->error = 0;
			return S_OK;
		}
		catch (CBaseErr *e)
		{
			// check for lost database connection; if yes, component is toast
			if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
				((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
				m_bCanBePooled = FALSE;

			pData->retval = e->ErrorType();
			pData->error = e->ErrorNum();
			delete e;
			return E_FAIL;
		}
		catch (...)
		{
			WriteMessageToEventLog(TEXT("Unhandled exception."));
			pData->retval = ERR_TYPE_LOGIC;
			pData->error = 0;
			m_bCanBePooled = FALSE;
			return E_FAIL;
		}
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in, VARIANT* txn_out)
{
		PORDER_STATUS_DATA	pOrderStatus;
		COM_DATA				*pData;
		try
		{
			pData = (COM_DATA*)txn_in.parray->pvData;
			pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

			memcpy(pOrderStatus, &pData->u.OrderStatus, sizeof(ORDER_STATUS_DATA));

			m_pTxn->OrderStatus();

			VariantInit(txn_out);
			txn_out->vt = VT_SAFEARRAY;
			txn_out->parray = SafeArrayCreateVector( VT_UI1,

			txn_in.parray->rgsabound->cElements,

			txn_in.parray->rgsabound->cElements);
			pData = (COM_DATA*)txn_out->parray->pvData;

			memcpy( &pData->u.OrderStatus, pOrderStatus, sizeof(ORDER_STATUS_DATA));

			pData->retval = ERR_SUCCESS;
			pData->error = 0;
			return S_OK;
		}
		catch (CBaseErr *e)
		{
			// check for lost database connection; if yes, component is toast
			if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
				((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
				m_bCanBePooled = FALSE;

			pData->retval = e->ErrorType();
			pData->error = e->ErrorNum();
			delete e;
			return E_FAIL;
		}
		catch (...)
		{
			WriteMessageToEventLog(TEXT("Unhandled exception."));
			pData->retval = ERR_TYPE_LOGIC;
			pData->error = 0;
			m_bCanBePooled = FALSE;
			return E_FAIL;
		}
}
```

\tpcc_com_all.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
```

```
#pragma once
#endif

/* Forward Declarations */

#ifndef __TPCC_FWD_DEFINED__
#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif          /* __TPCC_FWD_DEFINED__ */


#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif          /* __NewOrder_FWD_DEFINED__ */


#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif          /* __OrderStatus_FWD_DEFINED__ */


#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif          /* __Payment_FWD_DEFINED__ */


#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif          /* __StockLevel_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */




extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif
EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

\tpcc_com_all\src\tpcc_com_all.idl

```
/*          FILE:                   TPCC.IDL
```

```
 *                                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                                      Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                                      not yet audited
 *
 *              PURPOSE:    IDL source for TPCC.dll.  This file is
processed by the MIDL tool to
 *                                                      produce the type
library (TPCC.tlb) and marshalling code.
 *
 *  Change history:
 *                              4.20.000 - first version
 */

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;


import "oaidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

[
        uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
        version(1.0),
        helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
        importlib("stdole32.tlb");
        importlib("stdole2.tlb");

        [
                uuid(122A3128-2520-11D3-BA71-
00C04FBFE08B),
                helpstring("All Txns Class")
        ]
        coclass TPCC
        {
                [default] interface ITPCC;
        };


        [
                uuid(975BAABF-84A7-11D2-BA47-
00C04FBFE08B),
        ]       helpstring("NewOrder Class")
        coclass NewOrder
        {
                [default] interface ITPCC;
        };


        [
                uuid(266836AD-A50D-11D2-BA4E-
00C04FBFE08B),
                helpstring("OrderStatus Class")
        ]
        coclass OrderStatus
        {
                [default] interface ITPCC;
        };


        [
                uuid(CD02F7EF-A4FA-11D2-BA4E-
00C04FBFE08B),
                helpstring("Payment Class")
        ]
        coclass Payment
        {
                [default] interface ITPCC;
        };


        [
                uuid(2668369E-A50D-11D2-BA4E-
00C04FBFE08B),
                helpstring("StockLevel Class")
        ]
        coclass StockLevel
        {
                [default] interface ITPCC;
        };

};
```

\tpcc_com_all\src\tpcc_com_all.rc

```
//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS

//////////////////////////////////////////////////////////////////////////////
// Generated from the TEXTINCLUDE 2 resource.
//
#include "winres.h"

//////////////////////////////////////////////////////////////////////////////
#undef APSTUDIO_READONLY_SYMBOLS

//////////////////////////////////////////////////////////////////////////////
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif //_WIN32

#ifdef APSTUDIO_INVOKED
//////////////////////////////////////////////////////////////////////////////
//
// TEXTINCLUDE
//

1 TEXTINCLUDE DISCARDABLE
BEGIN
    "resource.h\0"
END

2 TEXTINCLUDE DISCARDABLE
BEGIN
    "#include ""winres.h""\r\n"
    "\0"
END

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
    "\0"
END

#endif    // APSTUDIO_INVOKED


#ifndef _MAC
//////////////////////////////////////////////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
 FILEVERSION 1,0,0,1
 PRODUCTVERSION 1,0,0,1
 FILEFLAGSMASK 0x3fL
```

```
#ifdef _DEBUG
 FILEFLAGS 0x1L
#else
 FILEFLAGS 0x0L
#endif
 FILEOS 0x4L
 FILETYPE 0x2L
 FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


/////////////////////////////////////////////////////////////////////////////
//
// REGISTRY
//

IDR_TPCC                REGISTRY DISCARDABLE    "tpcc_com_all.rgs"
IDR_NEWORDER            REGISTRY DISCARDABLE
"tpcc_com_no.rgs"
IDR_ORDERSTATUS         REGISTRY DISCARDABLE
"tpcc_com_os.rgs"
IDR_PAYMENT             REGISTRY DISCARDABLE    "tpcc_com_pay.rgs"
IDR_STOCKLEVEL          REGISTRY DISCARDABLE    "tpcc_com_sl.rgs"

/////////////////////////////////////////////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME            "tpcc_com_all"
END

#endif    // English (U.S.) resources
/////////////////////////////////////////////////////////////////////////////



#ifndef APSTUDIO_INVOKED
/////////////////////////////////////////////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

/////////////////////////////////////////////////////////////////////////////
#endif    // not APSTUDIO_INVOKED
```

\tpcc_com_all\src\tpcc_com_all.rgs

```
HKCR
{
        TPCC.AllTxns.1 = s 'All Txns Class'
        {
                CLSID = s '{122A3128-2520-11D3-BA71-
00C04FBFE08B}'
        }
        TPCC.AllTxns = s 'TPCC Class'
        {
                CurVer = s 'TPCC.AllTxns.1'
        }
        NoRemove CLSID
        {
                ForceRemove {122A3128-2520-11D3-BA71-
00C04FBFE08B} = s 'TPCC Class'
                {
                        ProgID = s 'TPCC.AllTxns.1'
                        VersionIndependentProgID = s
'TPCC.AllTxns'
                        InprocServer32 = s
'%MODULE%'
                        {
                                val
ThreadingModel = s 'Both'
                        }
                }
        }
}
```

\tpcc_com_all\src\tpcc_com_all_i.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_
```

```
#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F
,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0
xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,
0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x
4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B);
#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf, W1, Zp8, env=Win64 (32b run,appending)
    protocol : dce , ms_ext, c_ext, robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if defined(_M_IA64) || defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif
```

```
#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F
,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0xC0,0x4F,0
xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,
0x4F,0xBF,0xE0,0x8B);


MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0xC0,0x
4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* defined(_M_IA64) || defined(_M_AMD64)*/
```

\tpcc_com_all\src\tpcc_com_no.rgs

```
HKCR
{
	TPCC.NewOrder.1 = s 'NewOrder Class'
	{
		CLSID = s '{975BAABF-84A7-11D2-BA47-
00C04FBFE08B}'
	}
	TPCC.NewOrder = s 'NewOrder Class'
	{
		CurVer = s 'TPCC.NewOrder.1'
	}
	NoRemove CLSID
	{
		ForceRemove {975BAABF-84A7-11D2-BA47-
00C04FBFE08B} = s 'NewOrder Class'
		{
			ProgID = s 'TPCC.NewOrder.1'
			VersionIndependentProgID = s
'TPCC.NewOrder'
			InprocServer32 = s
'%MODULE%'
			{
				val
ThreadingModel = s 'Both'
			}
		}
	}
}
```

\tpcc_com_all\src\tpcc_com_os.rgs

```
HKCR
{
	TPCC.OrderStatus.1 = s 'OrderStatus Class'
	{
		CLSID = s '{266836AD-A50D-11D2-BA4E-
00C04FBFE08B}'
	}
	TPCC.OrderStatus = s 'OrderStatus Class'
	{
		CurVer = s 'TPCC.OrderStatus.1'
	}
	NoRemove CLSID
	{
		ForceRemove {266836AD-A50D-11D2-BA4E-
00C04FBFE08B} = s 'OrderStatus Class'
		{
			ProgID = s
'TPCC.OrderStatus.1'
			VersionIndependentProgID = s
'TPCC.OrderStatus'
			InprocServer32 = s
'%MODULE%'
			{
				val
ThreadingModel = s 'Both'
			}
		}
	}
}
```

\tpcc_com_all\src\tpcc_com_pay.rgs

```
HKCR
{
	TPCC.Payment.1 = s 'Payment Class'
	{
		CLSID = s '{CD02F7EF-A4FA-11D2-BA4E-
00C04FBFE08B}'
	}
	TPCC.Payment = s 'Payment Class'
	{
		CurVer = s 'TPCC.Payment.1'
	}
	NoRemove CLSID
	{
		ForceRemove {CD02F7EF-A4FA-11D2-BA4E-
00C04FBFE08B} = s 'Payment Class'
		{
			ProgID = s 'TPCC.Payment.1'
			VersionIndependentProgID = s
'TPCC.Payment'
			InprocServer32 = s
'%MODULE%'
			{
				val
ThreadingModel = s 'Both'
			}
		}
	}
}
```

\tpcc_com_all\src\tpcc_com_ps.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the
interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to compile this
file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif
```

```
#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif          /* __ITPCC_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall Payment(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall Delivery(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall StockLevel(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall OrderStatus(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else           /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
            ITPCC * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void **ppvObject);

        ULONG ( STDMETHODCALLTYPE *AddRef )(
            ITPCC * This);

        ULONG ( STDMETHODCALLTYPE *Release )(
            ITPCC * This);

        HRESULT ( __stdcall *NewOrder )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);
        HRESULT ( __stdcall *Payment )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);

        HRESULT ( __stdcall *Delivery )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);

        HRESULT ( __stdcall *StockLevel )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);

        HRESULT ( __stdcall *OrderStatus )(
            ITPCC * This,
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out);

        HRESULT ( __stdcall *CallSetComplete )(
            ITPCC * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl *lpVtbl;
    };



#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)    \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This)                \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)               \
    (This)->lpVtbl -> Release(This)


#define ITPCC_NewOrder(This,txn_in,txn_out)          \
    (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out)           \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)
```

```
#define ITPCC_Delivery(This,txn_in,txn_out)            \
    (This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out)          \
    (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out)         \
    (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This)    \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */


#endif         /* C style interface */



HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);



#endif         /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(     unsigned
long *, unsigned long            , VARIANT * );
unsigned char * __RPC_USER  VARIANT_UserMarshal(  unsigned long
*, unsigned char *, VARIANT * );
unsigned char * __RPC_USER  VARIANT_UserUnmarshal(unsigned long
*, unsigned char *, VARIANT * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long
*, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

\tpcc_com_all\src\tpcc_com_sl.rgs

```
HKCR
{
        TPCC.StockLevel.1 = s 'StockLevel Class'
        {
                CLSID = s '{2668369E-A50D-11D2-BA4E-
00C04FBFE08B}'
        }
        TPCC.StockLevel = s 'StockLevel Class'
        {
                CurVer = s 'TPCC.StockLevel.1'
        }
        NoRemove CLSID
        {
                ForceRemove {2668369E-A50D-11D2-BA4E-
00C04FBFE08B} = s 'StockLevel Class'
                {
                        ProgID = s 'TPCC.StockLevel.1'
                        VersionIndependentProgID = s
'TPCC.StockLevel'
                        InprocServer32 = s
'%MODULE%'
                        {
                                val
ThreadingModel = s 'Both'
                        }
                }
        }
}
```

\tpcc_com_ps\src\dlldata.c

```
/*********************************************************
   DllData file -- generated by MIDL compiler

        DO NOT ALTER THIS FILE

   This file is regenerated by MIDL on every IDL file compile.

   To completely reconstruct this file, delete it and rerun MIDL
   on all the IDL files in this DLL, specifying this file for the
   /dlldata command line option

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


PROXYFILE_LIST_START
/* Start of list */
  REFERENCE_PROXY_FILE( tpcc_com_ps ),
/* End of list */
PROXYFILE_LIST_END


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

#ifdef __cplusplus
}  /*extern "C" */
#endif

/* end of generated dlldata file */
```

tpcc_com_ps\src\tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
                DllGetClassObject       @1  PRIVATE
                DllCanUnloadNow         @2              PRIVATE
                DllRegisterServer       @3  PRIVATE     @4
                GetProxyDllInfo
PRIVATE
                DllUnregisterServer                     @5
PRIVATE
```

tpcc_com_ps\src\tpcc_com_ps.h

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the
interfaces */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


/* verify that the <rpcndr.h> version is high enough to compile this
file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#ifndef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifndef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__
#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif         /* __ITPCC_FWD_DEFINED__ */


/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */


EXTERN_C const IID IID_ITPCC;

#if defined(__cplusplus) && !defined(CINTERFACE)

    MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT __stdcall NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

        virtual HRESULT __stdcall Payment(
            /* [in] */ VARIANT txn_in,
```

```
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Delivery(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall StockLevel(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall OrderStatus(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall CallSetComplete( void) = 0;

  };

#else           /* C style interface */

  typedef struct ITPCCVtbl
  {
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
        ITPCC * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef )(
        ITPCC * This);

    ULONG ( STDMETHODCALLTYPE *Release )(
        ITPCC * This);

    HRESULT ( __stdcall *NewOrder )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *Payment )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *Delivery )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *StockLevel )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *OrderStatus )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *CallSetComplete )(
        ITPCC * This);

    END_INTERFACE
  } ITPCCVtbl;

  interface ITPCC
  {
    CONST_VTBL struct ITPCCVtbl *lpVtbl;
  };


#ifdef COBJMACROS


#define ITPCC_QueryInterface(This,riid,ppvObject)    \
  (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This)              \
  (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This)             \
  (This)->lpVtbl -> Release(This)


#define ITPCC_NewOrder(This,txn_in,txn_out)         \
  (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out)          \
  (This)->lpVtbl -> Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out)         \
  (This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out)       \
  (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out)      \
  (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This)    \
  (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */

#endif          /* C style interface */


HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
```

```
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);


void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);



#endif          /* __ITPCC_INTERFACE_DEFINED__ */


/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER  VARIANT_UserSize(     unsigned
long *, unsigned long            , VARIANT * );
unsigned char * __RPC_USER  VARIANT_UserMarshal(  unsigned long
*, unsigned char *, VARIANT * );
unsigned char * __RPC_USER  VARIANT_UserUnmarshal(unsigned long
*, unsigned char *, VARIANT * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long
*, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps\src\tpcc_com_ps.idl

```
                FILE:                   ITPCC.IDL
 *                                                      Microsoft TPC-C
Kit Ver. 4.20.000
 *                                                      Copyright
Microsoft, 1999
 *              All Rights Reserved
 *
 *                                                      not yet audited
 *
 *              PURPOSE:    Defines the interface used by TPCC. This
interface can be implemented by C++ components.
 *
 *  Change history:
 *                          4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

                [
                        object,
                        oleautomation,
                        uuid(FEEE6AA2-84B1-11d2-BA47-
00C04FBFE08B),
                        helpstring("ITPCC Interface"),
                        pointer_default(unique)
                ]
                interface ITPCC : IUnknown
                {

        HRESULT _stdcall NewOrder

                (

                [in] VARIANT  txn_in,

                [out] VARIANT *txn_out

                );

        HRESULT _stdcall Payment

                (

                [in] VARIANT txn_in,

                [out] VARIANT *txn_out

                );
        HRESULT _stdcall Delivery

                (

                [in] VARIANT txn_in,

                [out] VARIANT *txn_out

                );

        HRESULT _stdcall StockLevel

                (

                [in] VARIANT txn_in,

                [out] VARIANT *txn_out

                );

        HRESULT _stdcall OrderStatus

                (

                [in] VARIANT txn_in,

                [out] VARIANT *txn_out

                );

        HRESULT _stdcall CallSetComplete

                (

                );

};              // interface ITPCC
```

tpcc_com_ps\src\tpcc_com_ps_i.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */
```

```
/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif


#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xB
F,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win64 (32b run,appending)
    protocol : dce , ms_ext, c_ext, robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if defined(_M_IA64) || defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8) \
        const type name = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xB
F,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif
```

```
#endif /* defined(_M_IA64) || defined(_M_AMD64)*/
```

tpcc_com_ps\src\tpcc_com_ps_p.c

```
#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )

#if !defined(_M_IA64) && !defined(_M_AMD64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile this
file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   1023
#define PROC_FORMAT_STRING_SIZE   193
#define WIRE_MARSHAL_TABLE_SIZE   01
#define TRANSMIT_AS_TABLE_SIZE 0

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


static RPC_SYNTAX_IDENTIFIER  _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,
0x60}},{2,0}};


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;


extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN32__)
#error  Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it
uses these features:
#error   -Oif or -Oicf, [wire_marshal] or [user_marshal] attribute.
#error However, your C/C++ compilation flags indicate you intend to
run this app on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION
error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {
                        /* Procedure NewOrder */

                                        0x33,                   /*
FC_AUTO_HANDLE */
                                        0x6c,                   /*
Old Flags:  object, Oi2 */
/*  2 */        NdrFcLong( 0x0 ),       /* 0 */
/*  6 */        NdrFcShort( 0x3 ),      /* 3 */
/*  8 */        NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28
*/
/* 10 */        NdrFcShort( 0x0 ),      /* 0 */
/* 12 */        NdrFcShort( 0x8 ),      /* 8 */
/* 14 */        0x7,                    /* Oi2 Flags:  srv must size, clt
must size, has return, */
                                        0x3,                    /* 3
*/

                /* Parameter txn_in */

/* 16 */        NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 18 */        NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 20 */        NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

                /* Parameter txn_out */

/* 22 */        NdrFcShort( 0x4113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=16 */
/* 24 */        NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20
*/
/* 26 */        NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

                /* Return value */

/* 28 */        NdrFcShort( 0x70 ),     /* Flags:  out, return, base
type, */
/* 30 */        NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24
*/
/* 32 */        0x8,                    /* FC_LONG */
                                        0x0,                    /* 0
*/

                /* Procedure Payment */

/* 34 */        0x33,                   /* FC_AUTO_HANDLE */
                                        0x6c,                   /*
Old Flags:  object, Oi2 */
/* 36 */        NdrFcLong( 0x0 ),       /* 0 */
/* 40 */        NdrFcShort( 0x4 ),      /* 4 */
/* 42 */        NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28
*/
/* 44 */        NdrFcShort( 0x0 ),      /* 0 */
/* 46 */        NdrFcShort( 0x8 ),      /* 8 */
```

```
/* 48 */    0x7,                    /* Oi2 Flags:  srv must size, clt
must size, has return, */
                                    0x3,                    /* 3
*/

            /* Parameter txn_in */

/* 50 */    NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 52 */    NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 54 */    NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

            /* Parameter txn_out */

/* 56 */    NdrFcShort( 0x4113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=16 */
/* 58 */    NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20
*/
/* 60 */    NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

            /* Return value */

/* 62 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base
type, */
/* 64 */    NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24
*/
/* 66 */    0x8,                    /* FC_LONG */
                                    0x0,                    /* 0
*/

            /* Procedure Delivery */

/* 68 */    0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /*
Old Flags:  object, Oi2 */
/* 70 */    NdrFcLong( 0x0 ),       /* 0 */
/* 74 */    NdrFcShort( 0x5 ),      /* 5 */
/* 76 */    NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28
*/
/* 78 */    NdrFcShort( 0x0 ),      /* 0 */
/* 80 */    NdrFcShort( 0x8 ),      /* 8 */
/* 82 */    0x7,                    /* Oi2 Flags:  srv must size, clt
must size, has return, */
                                    0x3,                    /* 3
*/

            /* Parameter txn_in */

/* 84 */    NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 86 */    NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 88 */    NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

            /* Parameter txn_out */

/* 90 */    NdrFcShort( 0x4113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=16 */
/* 92 */    NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20
*/
/* 94 */    NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

            /* Return value */

/* 96 */    NdrFcShort( 0x70 ),     /* Flags:  out, return, base
type, */
/* 98 */    NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24
*/
/* 100 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0
*/

            /* Procedure StockLevel */

/* 102 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /*
Old Flags:  object, Oi2 */
/* 104 */   NdrFcLong( 0x0 ),       /* 0 */
/* 108 */   NdrFcShort( 0x6 ),      /* 6 */
/* 110 */   NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28
*/
/* 112 */   NdrFcShort( 0x0 ),      /* 0 */
/* 114 */   NdrFcShort( 0x8 ),      /* 8 */
/* 116 */   0x7,                    /* Oi2 Flags:  srv must size, clt
must size, has return, */
                                    0x3,                    /* 3
*/

            /* Parameter txn_in */

/* 118 */   NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 120 */   NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 122 */   NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

            /* Parameter txn_out */

/* 124 */   NdrFcShort( 0x4113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=16 */
/* 126 */   NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20
*/
/* 128 */   NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

            /* Return value */

/* 130 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base
type, */
/* 132 */   NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24
*/
/* 134 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0
*/

            /* Procedure OrderStatus */

/* 136 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /*
Old Flags:  object, Oi2 */
/* 138 */   NdrFcLong( 0x0 ),       /* 0 */
/* 142 */   NdrFcShort( 0x7 ),      /* 7 */
/* 144 */   NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28
*/
/* 146 */   NdrFcShort( 0x0 ),      /* 0 */
/* 148 */   NdrFcShort( 0x8 ),      /* 8 */
/* 150 */   0x7,                    /* Oi2 Flags:  srv must size, clt
must size, has return, */
                                    0x3,                    /* 3
*/

            /* Parameter txn_in */

/* 152 */   NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 154 */   NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 156 */   NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

            /* Parameter txn_out */

/* 158 */   NdrFcShort( 0x4113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=16 */
/* 160 */   NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20
*/
/* 162 */   NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

            /* Return value */

/* 164 */   NdrFcShort( 0x70 ),     /* Flags:  out, return, base
type, */
/* 166 */   NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24
*/
/* 168 */   0x8,                    /* FC_LONG */
                                    0x0,                    /* 0
*/

            /* Procedure CallSetComplete */

/* 170 */   0x33,                   /* FC_AUTO_HANDLE */
                                    0x6c,                   /*
Old Flags:  object, Oi2 */
/* 172 */   NdrFcLong( 0x0 ),       /* 0 */
/* 176 */   NdrFcShort( 0x8 ),      /* 8 */
```

```
/* 178 */      NdrFcShort( 0x8 ),          /* x86 Stack size/offset = 8 */
/* 180 */      NdrFcShort( 0x0 ),          /* 0 */
/* 182 */      NdrFcShort( 0x8 ),          /* 8 */
/* 184 */      0x4,                        /* Oi2 Flags:  has return, */
                                           0x1,                        /* 1
*/

               /* Return value */

/* 186 */      NdrFcShort( 0x70 ),         /* Flags:  out, return, base
type, */
/* 188 */      NdrFcShort( 0x4 ),          /* x86 Stack size/offset = 4 */
/* 190 */      0x8,                        /* FC_LONG */
                                           0x0,                        /* 0
*/

                                           0x0

       }
   };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
   {
      0,
      {
                                           NdrFcShort( 0x0 ),          /* 0
*/
/*  2 */
                                           0x12, 0x0,    /* FC_UP */
/*  4 */       NdrFcShort( 0x3ca ),        /* Offset= 970 (974) */
/*  6 */
                                           0x2b,                       /*
FC_NON_ENCAPSULATED_UNION */
                                           0x9,                        /*
FC_ULONG */
/*  8 */       0x7,                        /* Corr desc: FC_USHORT */
                                           0x0,                        /*
*/
/* 10 */       NdrFcShort( 0xfff8 ),       /* -8 */
/* 12 */       NdrFcShort( 0x2 ),          /* Offset= 2 (14) */
/* 14 */       NdrFcShort( 0x10 ),         /* 16 */
/* 16 */       NdrFcShort( 0x2f ),         /* 47 */
/* 18 */       NdrFcLong( 0x14 ),          /* 20 */
/* 22 */       NdrFcShort( 0x800b ),       /* Simple arm type: FC_HYPER
*/
/* 24 */       NdrFcLong( 0x3 ),           /* 3 */
/* 28 */       NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG
*/
/* 30 */       NdrFcLong( 0x11 ),          /* 17 */
/* 34 */       NdrFcShort( 0x8001 ),       /* Simple arm type: FC_BYTE
*/
/* 36 */       NdrFcLong( 0x2 ),           /* 2 */
/* 40 */       NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT
*/
/* 42 */       NdrFcLong( 0x4 ),           /* 4 */
/* 46 */       NdrFcShort( 0x800a ),       /* Simple arm type: FC_FLOAT
*/
/* 48 */       NdrFcLong( 0x5 ),           /* 5 */
/* 52 */       NdrFcShort( 0x800c ),       /* Simple arm type:
FC_DOUBLE */
/* 54 */       NdrFcLong( 0xb ),           /* 11 */
/* 58 */       NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT
*/
/* 60 */       NdrFcLong( 0xa ),           /* 10 */
/* 64 */       NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG
*/
/* 66 */       NdrFcLong( 0x6 ),           /* 6 */
/* 70 */       NdrFcShort( 0xe8 ),         /* Offset= 232 (302) */
/* 72 */       NdrFcLong( 0x7 ),           /* 7 */
/* 76 */       NdrFcShort( 0x800c ),       /* Simple arm type:
FC_DOUBLE */
/* 78 */       NdrFcLong( 0x8 ),           /* 8 */
/* 82 */       NdrFcShort( 0xe2 ),         /* Offset= 226 (308) */
/* 84 */       NdrFcLong( 0xd ),           /* 13 */
/* 88 */       NdrFcShort( 0xf4 ),         /* Offset= 244 (332) */
/* 90 */       NdrFcLong( 0x9 ),           /* 9 */
/* 94 */       NdrFcShort( 0x100 ),        /* Offset= 256 (350) */
/* 96 */       NdrFcLong( 0x2000 ),        /* 8192 */
/* 100 */      NdrFcShort( 0x10c ),        /* Offset= 268 (368) */
/* 102 */      NdrFcLong( 0x24 ),          /* 36 */
/* 106 */      NdrFcShort( 0x31a ),        /* Offset= 794 (900) */
/* 108 */      NdrFcLong( 0x4024 ),        /* 16420 */
/* 112 */      NdrFcShort( 0x314 ),        /* Offset= 788 (900) */
/* 114 */      NdrFcLong( 0x4011 ),        /* 16401 */
/* 118 */      NdrFcShort( 0x312 ),        /* Offset= 786 (904) */
/* 120 */      NdrFcLong( 0x4002 ),        /* 16386 */
/* 124 */      NdrFcShort( 0x310 ),        /* Offset= 784 (908) */
/* 126 */      NdrFcLong( 0x4003 ),        /* 16387 */
/* 130 */      NdrFcShort( 0x30e ),        /* Offset= 782 (912) */
/* 132 */      NdrFcLong( 0x4014 ),        /* 16404 */
/* 136 */      NdrFcShort( 0x30c ),        /* Offset= 780 (916) */
/* 138 */      NdrFcLong( 0x4004 ),        /* 16388 */
/* 142 */      NdrFcShort( 0x30a ),        /* Offset= 778 (920) */
/* 144 */      NdrFcLong( 0x4005 ),        /* 16389 */
/* 148 */      NdrFcShort( 0x308 ),        /* Offset= 776 (924) */
/* 150 */      NdrFcLong( 0x400b ),        /* 16395 */
/* 154 */      NdrFcShort( 0x2f2 ),        /* Offset= 754 (908) */
/* 156 */      NdrFcLong( 0x400a ),        /* 16394 */
/* 160 */      NdrFcShort( 0x2f0 ),        /* Offset= 752 (912) */
/* 162 */      NdrFcLong( 0x4006 ),        /* 16390 */
/* 166 */      NdrFcShort( 0x2fa ),        /* Offset= 762 (928) */
/* 168 */      NdrFcLong( 0x4007 ),        /* 16391 */
/* 172 */      NdrFcShort( 0x2f0 ),        /* Offset= 752 (924) */
/* 174 */      NdrFcLong( 0x4008 ),        /* 16392 */
/* 178 */      NdrFcShort( 0x2f2 ),        /* Offset= 754 (932) */
/* 180 */      NdrFcLong( 0x400d ),        /* 16397 */
/* 184 */      NdrFcShort( 0x2f0 ),        /* Offset= 752 (936) */
/* 186 */      NdrFcLong( 0x4009 ),        /* 16393 */
/* 190 */      NdrFcShort( 0x2ee ),        /* Offset= 750 (940) */
/* 192 */      NdrFcLong( 0x6000 ),        /* 24576 */
/* 196 */      NdrFcShort( 0x2ec ),        /* Offset= 748 (944) */
/* 198 */      NdrFcLong( 0x400c ),        /* 16396 */
/* 202 */      NdrFcShort( 0x2ea ),        /* Offset= 746 (948) */
/* 204 */      NdrFcLong( 0x10 ),          /* 16 */
/* 208 */      NdrFcShort( 0x8002 ),       /* Simple arm type: FC_CHAR
*/
/* 210 */      NdrFcLong( 0x12 ),          /* 18 */
/* 214 */      NdrFcShort( 0x8006 ),       /* Simple arm type: FC_SHORT
*/
/* 216 */      NdrFcLong( 0x13 ),          /* 19 */
/* 220 */      NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG
*/
/* 222 */      NdrFcLong( 0x15 ),          /* 21 */
/* 226 */      NdrFcShort( 0x800b ),       /* Simple arm type: FC_HYPER
*/
/* 228 */      NdrFcLong( 0x16 ),          /* 22 */
/* 232 */      NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG
*/
/* 234 */      NdrFcLong( 0x17 ),          /* 23 */
/* 238 */      NdrFcShort( 0x8008 ),       /* Simple arm type: FC_LONG
*/
/* 240 */      NdrFcLong( 0xe ),           /* 14 */
/* 244 */      NdrFcShort( 0x2c8 ),        /* Offset= 712 (956) */
/* 246 */      NdrFcLong( 0x400e ),        /* 16398 */
/* 250 */      NdrFcShort( 0x2cc ),        /* Offset= 716 (966) */
/* 252 */      NdrFcLong( 0x4010 ),        /* 16400 */
/* 256 */      NdrFcShort( 0x2ca ),        /* Offset= 714 (970) */
/* 258 */      NdrFcLong( 0x4012 ),        /* 16402 */
/* 262 */      NdrFcShort( 0x286 ),        /* Offset= 646 (908) */
/* 264 */      NdrFcLong( 0x4013 ),        /* 16403 */
/* 268 */      NdrFcShort( 0x284 ),        /* Offset= 644 (912) */
/* 270 */      NdrFcLong( 0x4015 ),        /* 16405 */
/* 274 */      NdrFcShort( 0x282 ),        /* Offset= 642 (916) */
/* 276 */      NdrFcLong( 0x4016 ),        /* 16406 */
/* 280 */      NdrFcShort( 0x278 ),        /* Offset= 632 (912) */
/* 282 */      NdrFcLong( 0x4017 ),        /* 16407 */
/* 286 */      NdrFcShort( 0x272 ),        /* Offset= 626 (912) */
/* 288 */      NdrFcLong( 0x0 ),           /* 0 */
/* 292 */      NdrFcShort( 0x0 ),          /* Offset= 0 (292) */
/* 294 */      NdrFcLong( 0x1 ),           /* 1 */
/* 298 */      NdrFcShort( 0x0 ),          /* Offset= 0 (298) */
/* 300 */      NdrFcShort( 0xffffffff ),   /* Offset= -1 (299) */
/* 302 */
                                           0x15,                       /*
FC_STRUCT */
                                           0x7,                        /* 7
*/
/* 304 */      NdrFcShort( 0x8 ),          /* 8 */
/* 306 */      0xb,                        /* FC_HYPER */
```

```
                                0x5b,                   /*
FC_END */
/* 308 */

                                0x12, 0x0,    /* FC_UP */
/* 310 */      NdrFcShort( 0xc ),      /* Offset= 12 (322) */
/* 312 */

                                0x1b,                   /*
FC_CARRAY */

                                0x1,                    /* 1
*/
/* 314 */      NdrFcShort( 0x2 ),      /* 2 */
/* 316 */      0x9,                    /* Corr desc: FC_ULONG */
                                0x0,                    /*
*/
/* 318 */      NdrFcShort( 0xfffc ),   /* -4 */
/* 320 */      0x6,                    /* FC_SHORT */
                                0x5b,                   /*
FC_END */
/* 322 */

                                0x17,                   /*
FC_CSTRUCT */

                                0x3,                    /* 3
*/
/* 324 */      NdrFcShort( 0x8 ),      /* 8 */
/* 326 */      NdrFcShort( 0xffffffd2 ),   /* Offset= -14 (312) */
/* 328 */      0x8,                    /* FC_LONG */
                                0x8,                    /*
FC_LONG */
/* 330 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 332 */

                                0x2f,                   /*
FC_IP */

                                0x5a,                   /*
FC_CONSTANT_IID */
/* 334 */      NdrFcLong( 0x0 ),       /* 0 */
/* 338 */      NdrFcShort( 0x0 ),      /* 0 */
/* 340 */      NdrFcShort( 0x0 ),      /* 0 */
/* 342 */      0xc0,                   /* 192 */
                                0x0,                    /* 0
*/
/* 344 */      0x0,                    /* 0 */
                                0x0,                    /* 0
*/
/* 346 */      0x0,                    /* 0 */
                                0x0,                    /* 0
*/
/* 348 */      0x0,                    /* 0 */
                                0x46,                   /*
70 */
/* 350 */
```

```
                                0x2f,                   /*
FC_IP */                        0x5a,                   /*
FC_CONSTANT_IID */
/* 352 */      NdrFcLong( 0x20400 ),   /* 132096 */
/* 356 */      NdrFcShort( 0x0 ),      /* 0 */
/* 358 */      NdrFcShort( 0x0 ),      /* 0 */
/* 360 */      0xc0,                   /* 192 */
                                0x0,                    /* 0
*/
/* 362 */      0x0,                    /* 0 */
                                0x0,                    /* 0
*/
/* 364 */      0x0,                    /* 0 */
                                0x0,                    /* 0
*/
/* 366 */      0x0,                    /* 0 */
                                0x46,                   /*
70 */
/* 368 */

                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 370 */      NdrFcShort( 0x2 ),      /* Offset= 2 (372) */
/* 372 */

                                0x12, 0x0,    /* FC_UP */
/* 374 */      NdrFcShort( 0x1fc ),    /* Offset= 508 (882) */
/* 376 */

                                0x2a,                   /*
FC_ENCAPSULATED_UNION */

                                0x49,                   /*
73 */
/* 378 */      NdrFcShort( 0x18 ),     /* 24 */
/* 380 */      NdrFcShort( 0xa ),      /* 10 */
/* 382 */      NdrFcLong( 0x8 ),       /* 8 */
/* 386 */      NdrFcShort( 0x58 ),     /* Offset= 88 (474) */
/* 388 */      NdrFcLong( 0xd ),       /* 13 */
/* 392 */      NdrFcShort( 0x78 ),     /* Offset= 120 (512) */
/* 394 */      NdrFcLong( 0x9 ),       /* 9 */
/* 398 */      NdrFcShort( 0x94 ),     /* Offset= 148 (546) */
/* 400 */      NdrFcLong( 0xc ),       /* 12 */
/* 404 */      NdrFcShort( 0xbc ),     /* Offset= 188 (592) */
/* 406 */      NdrFcLong( 0x24 ),      /* 36 */
/* 410 */      NdrFcShort( 0x114 ),    /* Offset= 276 (686) */
/* 412 */      NdrFcLong( 0x800d ),    /* 32781 */
/* 416 */      NdrFcShort( 0x130 ),    /* Offset= 304 (720) */
/* 418 */      NdrFcLong( 0x10 ),      /* 16 */
/* 422 */      NdrFcShort( 0x148 ),    /* Offset= 328 (750) */
/* 424 */      NdrFcLong( 0x2 ),       /* 2 */
/* 428 */      NdrFcShort( 0x160 ),    /* Offset= 352 (780) */
/* 430 */      NdrFcLong( 0x3 ),       /* 3 */
/* 434 */      NdrFcShort( 0x178 ),    /* Offset= 376 (810) */
/* 436 */      NdrFcLong( 0x14 ),      /* 20 */
/* 440 */      NdrFcShort( 0x190 ),    /* Offset= 400 (840) */
/* 442 */      NdrFcShort( 0xffffffff ),   /* Offset= -1 (441) */
```

```
/* 444 */                       0x1b,                   /*
FC_CARRAY */
                                0x3,                    /* 3
*/
/* 446 */      NdrFcShort( 0x4 ),      /* 4 */
/* 448 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 450 */      NdrFcShort( 0x0 ),      /* 0 */
/* 452 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 454 */
                                0x48,                   /*
FC_VARIABLE_REPEAT */
                                0x49,                   /*
FC_FIXED_OFFSET */
/* 456 */      NdrFcShort( 0x4 ),      /* 4 */
/* 458 */      NdrFcShort( 0x0 ),      /* 0 */
/* 460 */      NdrFcShort( 0x1 ),      /* 1 */
/* 462 */      NdrFcShort( 0x0 ),      /* 0 */
/* 464 */      NdrFcShort( 0x0 ),      /* 0 */
/* 466 */      0x12, 0x0,    /* FC_UP */
/* 468 */      NdrFcShort( 0xffffff6e ),   /* Offset= -146 (322) */
/* 470 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 472 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 474 */

                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 476 */      NdrFcShort( 0x8 ),      /* 8 */
/* 478 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 480 */
                                0x46,                   /*
FC_NO_REPEAT */
                                0x5c,                   /*
FC_PAD */
/* 482 */      NdrFcShort( 0x4 ),      /* 4 */
```

```
/* 484 */      NdrFcShort( 0x4 ),          /* 4 */
/* 486 */      0x11, 0x0,    /* FC_RP */
/* 488 */      NdrFcShort( 0xfffffffd4 ),  /* Offset= -44 (444) */
/* 490 */
                                           0x5b,                       /*
FC_END */

                                           0x8,                        /*
FC_LONG */
/* 492 */      0x8,                        /* FC_LONG */
                                           0x5b,                       /*
FC_END */
/* 494 */
                                           0x21,                       /*
FC_BOGUS_ARRAY */
                                           0x3,                        /* 3
*/
/* 496 */      NdrFcShort( 0x0 ),          /* 0 */
/* 498 */      0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                           0x0,                        /*
*/
/* 500 */      NdrFcShort( 0x0 ),          /* 0 */
/* 502 */      NdrFcLong( 0xffffffff ),    /* -1 */
/* 506 */      0x4c,                       /* FC_EMBEDDED_COMPLEX
*/
                                           0x0,                        /* 0
*/
/* 508 */      NdrFcShort( 0xffffff50 ),   /* Offset= -176 (332) */
/* 510 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 512 */
                                           0x1a,                       /*
FC_BOGUS_STRUCT */
                                           0x3,                        /* 3
*/
/* 514 */      NdrFcShort( 0x8 ),          /* 8 */
/* 516 */      NdrFcShort( 0x0 ),          /* 0 */
/* 518 */      NdrFcShort( 0x6 ),          /* Offset= 6 (524) */
/* 520 */      0x8,                        /* FC_LONG */
                                           0x36,                       /*
FC_POINTER */
/* 522 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 524 */
                                           0x11, 0x0,    /* FC_RP */
/* 526 */      NdrFcShort( 0xfffffffe0 ),  /* Offset= -32 (494) */
/* 528 */
                                           0x21,                       /*
FC_BOGUS_ARRAY */
                                           0x3,                        /* 3
*/
/* 530 */      NdrFcShort( 0x0 ),          /* 0 */
/* 532 */      0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                           0x0,                        /*
*/
/* 534 */      NdrFcShort( 0x0 ),          /* 0 */
/* 536 */      NdrFcLong( 0xffffffff ),    /* -1 */
/* 540 */      0x4c,                       /* FC_EMBEDDED_COMPLEX
*/
                                           0x0,                        /* 0
*/
/* 542 */      NdrFcShort( 0xffffff40 ),   /* Offset= -192 (350) */
/* 544 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 546 */
                                           0x1a,                       /*
FC_BOGUS_STRUCT */
                                           0x3,                        /* 3
*/
/* 548 */      NdrFcShort( 0x8 ),          /* 8 */
/* 550 */      NdrFcShort( 0x0 ),          /* 0 */
/* 552 */      NdrFcShort( 0x6 ),          /* Offset= 6 (558) */
/* 554 */      0x8,                        /* FC_LONG */
                                           0x36,                       /*
FC_POINTER */
/* 556 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 558 */
                                           0x11, 0x0,    /* FC_RP */
/* 560 */      NdrFcShort( 0xfffffffe0 ),  /* Offset= -32 (528) */
/* 562 */
                                           0x1b,                       /*
FC_CARRAY */
                                           0x3,                        /* 3
*/
/* 564 */      NdrFcShort( 0x4 ),          /* 4 */
/* 566 */      0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                           0x0,                        /*
*/
/* 568 */      NdrFcShort( 0x0 ),          /* 0 */
/* 570 */
                                           0x4b,                       /*
FC_PP */
                                           0x5c,                       /*
FC_PAD */
/* 572 */
                                           0x48,                       /*
FC_VARIABLE_REPEAT */
                                           0x49,                       /*
FC_FIXED_OFFSET */
/* 574 */      NdrFcShort( 0x4 ),          /* 4 */
/* 576 */      NdrFcShort( 0x0 ),          /* 0 */
/* 578 */      NdrFcShort( 0x1 ),          /* 1 */
/* 580 */      NdrFcShort( 0x0 ),          /* 0 */
/* 582 */      NdrFcShort( 0x0 ),          /* 0 */
/* 584 */      0x12, 0x0,    /* FC_UP */
/* 586 */      NdrFcShort( 0x184 ),        /* Offset= 388 (974) */
/* 588 */
                                           0x5b,                       /*
FC_END */

                                           0x8,                        /*
FC_LONG */
/* 590 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 592 */
                                           0x1a,                       /*
FC_BOGUS_STRUCT */
                                           0x3,                        /* 3
*/
/* 594 */      NdrFcShort( 0x8 ),          /* 8 */
/* 596 */      NdrFcShort( 0x0 ),          /* 0 */
/* 598 */      NdrFcShort( 0x6 ),          /* Offset= 6 (604) */
/* 600 */      0x8,                        /* FC_LONG */
                                           0x36,                       /*
FC_POINTER */
/* 602 */      0x5c,                       /* FC_PAD */
                                           0x5b,                       /*
FC_END */
/* 604 */
                                           0x11, 0x0,    /* FC_RP */
/* 606 */      NdrFcShort( 0xfffffffd4 ),  /* Offset= -44 (562) */
/* 608 */
                                           0x2f,                       /*
FC_IP */
                                           0x5a,                       /*
FC_CONSTANT_IID */
/* 610 */      NdrFcLong( 0x2f ),          /* 47 */
/* 614 */      NdrFcShort( 0x0 ),          /* 0 */
/* 616 */      NdrFcShort( 0x0 ),          /* 0 */
/* 618 */      0xc0,                       /* 192 */
                                           0x0,                        /* 0
*/
/* 620 */      0x0,                        /* 0 */
                                           0x0,                        /* 0
*/
/* 622 */      0x0,                        /* 0 */
                                           0x0,                        /* 0
*/
/* 624 */      0x0,                        /* 0 */
```

```
                                0x46,                   /*
70 */
/* 626 */
                                0x1b,                   /*
FC_CARRAY */
                                0x0,                    /* 0
*/
/* 628 */      NdrFcShort( 0x1 ),      /* 1 */
/* 630 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 632 */      NdrFcShort( 0x4 ),      /* 4 */
/* 634 */      0x1,                    /* FC_BYTE */
                                0x5b,                   /*
FC_END */
/* 636 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3
*/
/* 638 */      NdrFcShort( 0x10 ),     /* 16 */
/* 640 */      NdrFcShort( 0x0 ),      /* 0 */
/* 642 */      NdrFcShort( 0xa ),      /* Offset= 10 (652) */
/* 644 */      0x8,                    /* FC_LONG */
                                0x8,                    /*
FC_LONG */
/* 646 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                0x0,                    /* 0
*/
/* 648 */      NdrFcShort( 0xffffffd8 ),       /* Offset= -40 (608) */
/* 650 */      0x36,                   /* FC_POINTER */
                                0x5b,                   /*
FC_END */
/* 652 */
                                0x12, 0x0,      /* FC_UP */
/* 654 */      NdrFcShort( 0xfffffffe4 ),      /* Offset= -28 (626) */
/* 656 */
                                0x1b,                   /*
FC_CARRAY */
                                0x3,                    /* 3
*/
/* 658 */      NdrFcShort( 0x4 ),      /* 4 */
/* 660 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 662 */      NdrFcShort( 0x0 ),      /* 0 */
/* 664 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
                                0x48,                   /*
FC_VARIABLE_REPEAT */
                                0x49,                   /*
FC_FIXED_OFFSET */
/* 668 */      NdrFcShort( 0x4 ),      /* 4 */
/* 670 */      NdrFcShort( 0x0 ),      /* 0 */
/* 672 */      NdrFcShort( 0x1 ),      /* 1 */
/* 674 */      NdrFcShort( 0x0 ),      /* 0 */
/* 676 */      NdrFcShort( 0x0 ),      /* 0 */
/* 678 */      0x12, 0x0,      /* FC_UP */
/* 680 */      NdrFcShort( 0xffffffd4 ),       /* Offset= -44 (636) */
/* 682 */
                                0x5b,                   /*
FC_END */


                                0x8,                    /*
FC_LONG */
/* 684 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 686 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3
*/
/* 688 */      NdrFcShort( 0x8 ),      /* 8 */
/* 690 */      NdrFcShort( 0x0 ),      /* 0 */
/* 692 */      NdrFcShort( 0x6 ),      /* Offset= 6 (698) */
/* 694 */      0x8,                    /* FC_LONG */
                                0x36,                   /*
FC_POINTER */
/* 696 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 698 */
                                0x11, 0x0,      /* FC_RP */
/* 700 */      NdrFcShort( 0xffffffd4 ),       /* Offset= -44 (656) */
/* 702 */
                                0x1d,                   /*
FC_SMFARRAY */
                                0x0,                    /* 0
*/
/* 704 */      NdrFcShort( 0x8 ),      /* 8 */
/* 706 */      0x1,                    /* FC_BYTE */
                                0x5b,                   /*
FC_END */
/* 708 */
                                0x15,                   /*
FC_STRUCT */
                                0x3,                    /* 3
*/
/* 710 */      NdrFcShort( 0x10 ),     /* 16 */
                                0x6,                    /*
FC_SHORT */
/* 712 */      0x6,                    /* FC_SHORT */
                                0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 714 */      0x0,                    /* 0 */
                                NdrFcShort( 0xffffffe8 ),       /*
Offset= -24 (690) */
                                0x5b,                   /*
FC_END */
/* 716 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3
*/
/* 718 */      NdrFcShort( 0x18 ),     /* 24 */
/* 720 */      NdrFcShort( 0x0 ),      /* 0 */
/* 722 */      NdrFcShort( 0xa ),      /* Offset= 10 (732) */
/* 724 */      0x8,                    /* FC_LONG */
                                0x36,                   /*
FC_POINTER */
/* 726 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                0x0,                    /* 0
*/
/* 728 */      NdrFcShort( 0xffffffe8 ),       /* Offset= -24 (704) */
/* 730 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 732 */
                                0x11, 0x0,      /* FC_RP */
/* 734 */      NdrFcShort( 0xffffff0c ),       /* Offset= -244 (490) */
/* 736 */
                                0x1b,                   /*
FC_CARRAY */
                                0x0,                    /* 0
*/
/* 738 */      NdrFcShort( 0x1 ),      /* 1 */
/* 740 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 742 */      NdrFcShort( 0x0 ),      /* 0 */
/* 744 */      0x1,                    /* FC_BYTE */
                                0x5b,                   /*
FC_END */
/* 746 */
                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 748 */      NdrFcShort( 0x8 ),      /* 8 */
/* 750 */      0x8,                    /* FC_LONG */
                                0x6,                    /*
FC_SHORT */
/* 752 */      0x6,                    /* FC_SHORT */
                                0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 754 */      0x0,                    /* 0 */
                                NdrFcShort( 0xffffffe1 ),       /*
Offset= -31 (724) */
                                0x5b,                   /*
FC_END */
/* 756 */
                                0x1a,                   /*
FC_BOGUS_STRUCT */
                                0x3,                    /* 3
*/
/* 758 */      NdrFcShort( 0x18 ),     /* 24 */
/* 760 */      NdrFcShort( 0x0 ),      /* 0 */
/* 762 */      NdrFcShort( 0xa ),      /* Offset= 10 (772) */
/* 764 */      0x8,                    /* FC_LONG */
                                0x36,                   /*
FC_POINTER */
/* 766 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                0x0,                    /* 0
*/
/* 768 */      NdrFcShort( 0xffffffe8 ),       /* Offset= -24 (744) */
/* 770 */      0x5c,                   /* FC_PAD */
                                0x5b,                   /*
FC_END */
/* 772 */
                                0x11, 0x0,      /* FC_RP */
/* 774 */      NdrFcShort( 0xffffff0c ),       /* Offset= -244 (530) */
/* 776 */
                                0x1b,                   /*
FC_CARRAY */
                                0x0,                    /* 0
*/
/* 778 */      NdrFcShort( 0x1 ),      /* 1 */
/* 780 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 782 */      NdrFcShort( 0x0 ),      /* 0 */
/* 784 */      0x1,                    /* FC_BYTE */
                                0x5b,                   /*
FC_END */
/* 786 */
                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 788 */      NdrFcShort( 0x8 ),      /* 8 */
```

```
/* 754 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 756 */
                                0x46,                   /*
FC_NO_REPEAT */
                                0x5c,                   /*
FC_PAD */
/* 758 */      NdrFcShort( 0x4 ),      /* 4 */
/* 760 */      NdrFcShort( 0x4 ),      /* 4 */
/* 762 */      0x12, 0x0,    /* FC_UP */
/* 764 */      NdrFcShort( 0xfffffff8 ),   /* Offset= -24 (740) */
/* 766 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 768 */      0x8,                    /* FC_LONG */
                                0x5b,                   /*
FC_END */
/* 770 */
                                0x1b,                   /*
FC_CARRAY */
                                0x1,                    /* 1
*/
/* 772 */      NdrFcShort( 0x2 ),      /* 2 */
/* 774 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 776 */      NdrFcShort( 0x0 ),      /* 0 */
/* 778 */      0x6,                    /* FC_SHORT */
                                0x5b,                   /*
FC_END */
/* 780 */
                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 782 */      NdrFcShort( 0x8 ),      /* 8 */
/* 784 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 786 */
                                0x46,                   /*
FC_NO_REPEAT */
```

```
                                0x5c,                   /*
FC_PAD */
/* 788 */      NdrFcShort( 0x4 ),      /* 4 */
/* 790 */      NdrFcShort( 0x4 ),      /* 4 */
/* 792 */      0x12, 0x0,    /* FC_UP */
/* 794 */      NdrFcShort( 0xfffffff8 ),   /* Offset= -24 (770) */
/* 796 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 798 */      0x8,                    /* FC_LONG */
                                0x5b,                   /*
FC_END */
/* 800 */
                                0x1b,                   /*
FC_CARRAY */
                                0x3,                    /* 3
*/
/* 802 */      NdrFcShort( 0x4 ),      /* 4 */
/* 804 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 806 */      NdrFcShort( 0x0 ),      /* 0 */
/* 808 */      0x8,                    /* FC_LONG */
                                0x5b,                   /*
FC_END */
/* 810 */
                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 812 */      NdrFcShort( 0x8 ),      /* 8 */
/* 814 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 816 */
                                0x46,                   /*
FC_NO_REPEAT */
                                0x5c,                   /*
FC_PAD */
/* 818 */      NdrFcShort( 0x4 ),      /* 4 */
/* 820 */      NdrFcShort( 0x4 ),      /* 4 */
/* 822 */      0x12, 0x0,    /* FC_UP */
/* 824 */      NdrFcShort( 0xfffffff8 ),   /* Offset= -24 (800) */
/* 826 */
                                0x5b,                   /*
FC_END */
```

```
                                0x8,                    /*
FC_LONG */
/* 828 */      0x8,                    /* FC_LONG */
                                0x5b,                   /*
FC_END */
/* 830 */
                                0x1b,                   /*
FC_CARRAY */
                                0x7,                    /* 7
*/
/* 832 */      NdrFcShort( 0x8 ),      /* 8 */
/* 834 */      0x19,                   /* Corr desc:  field pointer,
FC_ULONG */
                                0x0,                    /*
*/
/* 836 */      NdrFcShort( 0x0 ),      /* 0 */
/* 838 */      0xb,                    /* FC_HYPER */
                                0x5b,                   /*
FC_END */
/* 840 */
                                0x16,                   /*
FC_PSTRUCT */
                                0x3,                    /* 3
*/
/* 842 */      NdrFcShort( 0x8 ),      /* 8 */
/* 844 */
                                0x4b,                   /*
FC_PP */
                                0x5c,                   /*
FC_PAD */
/* 846 */
                                0x46,                   /*
FC_NO_REPEAT */
                                0x5c,                   /*
FC_PAD */
/* 848 */      NdrFcShort( 0x4 ),      /* 4 */
/* 850 */      NdrFcShort( 0x4 ),      /* 4 */
/* 852 */      0x12, 0x0,    /* FC_UP */
/* 854 */      NdrFcShort( 0xfffffff8 ),   /* Offset= -24 (830) */
/* 856 */
                                0x5b,                   /*
FC_END */

                                0x8,                    /*
FC_LONG */
/* 858 */      0x8,                    /* FC_LONG */
                                0x5b,                   /*
FC_END */
/* 860 */
                                0x15,                   /*
FC_STRUCT */
                                0x3,                    /* 3
*/
/* 862 */      NdrFcShort( 0x8 ),      /* 8 */
```

```
/* 864 */      0x8,                    /* FC_LONG */
                                       0x8,                    /*
FC_LONG */
/* 866 */      0x5c,                   /* FC_PAD */
                                       0x5b,                   /*
FC_END */
/* 868 */
                                       0x1b,                   /*
FC_CARRAY */
                                       0x3,                    /* 3
*/
/* 870 */      NdrFcShort( 0x8 ),      /* 8 */
/* 872 */      0x7,                    /* Corr desc: FC_USHORT */
                                       0x0,                    /*
*/
/* 874 */      NdrFcShort( 0xffd8 ),   /* -40 */
/* 876 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                       0x0,                    /* 0
*/
/* 878 */      NdrFcShort( 0xfffffffe ),  /* Offset= -18 (860) */
/* 880 */      0x5c,                   /* FC_PAD */
                                       0x5b,                   /*
FC_END */
/* 882 */
                                       0x1a,                   /*
FC_BOGUS_STRUCT */
                                       0x3,                    /* 3
*/
/* 884 */      NdrFcShort( 0x28 ),     /* 40 */
/* 886 */      NdrFcShort( 0xfffffffe ),  /* Offset= -18 (868) */
/* 888 */      NdrFcShort( 0x0 ),      /* Offset= 0 (888) */
/* 890 */      0x6,                    /* FC_SHORT */
                                       0x6,                    /*
FC_SHORT */
/* 892 */      0x8,                    /* FC_LONG */
                                       0x8,                    /*
FC_LONG */
/* 894 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                       0x0,                    /* 0
*/
/* 896 */      NdrFcShort( 0xfffffdf8 ),  /* Offset= -520 (376) */
/* 898 */      0x5c,                   /* FC_PAD */
                                       0x5b,                   /*
FC_END */
/* 900 */
                                       0x12, 0x0,    /* FC_UP */
/* 902 */      NdrFcShort( 0xfffffef6 ),  /* Offset= -266 (636) */
/* 904 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 906 */      0x1,                    /* FC_BYTE */           /*
FC_PAD */
/* 908 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 910 */      0x6,                    /* FC_SHORT */
                                       0x5c,                   /*
FC_PAD */
/* 912 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 914 */      0x8,                    /* FC_LONG */
                                       0x5c,                   /*
FC_PAD */
/* 916 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 918 */      0xb,                    /* FC_HYPER */
                                       0x5c,                   /*
FC_PAD */
/* 920 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 922 */      0xa,                    /* FC_FLOAT */
                                       0x5c,                   /*
FC_PAD */
/* 924 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 926 */      0xc,                    /* FC_DOUBLE */
                                       0x5c,                   /*
FC_PAD */
/* 928 */
                                       0x12, 0x0,    /* FC_UP */
/* 930 */      NdrFcShort( 0xfffffd8c ),  /* Offset= -628 (302) */
/* 932 */
                                       0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 934 */      NdrFcShort( 0xfffffd8e ),  /* Offset= -626 (308) */
/* 936 */
                                       0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 938 */      NdrFcShort( 0xfffffda2 ),  /* Offset= -606 (332) */
/* 940 */
                                       0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 942 */      NdrFcShort( 0xfffffdb0 ),  /* Offset= -592 (350) */
/* 944 */
                                       0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 946 */      NdrFcShort( 0xfffffdbe ),  /* Offset= -578 (368) */
/* 948 */
                                       0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 950 */      NdrFcShort( 0x2 ),      /* Offset= 2 (952) */
/* 952 */
                                       0x12, 0x0,    /* FC_UP */
/* 954 */      NdrFcShort( 0x14 ),     /* Offset= 20 (974) */
/* 956 */
                                       0x15,                   /*
FC_STRUCT */
                                       0x7,                    /* 7
*/
/* 958 */      NdrFcShort( 0x10 ),     /* 16 */
/* 960 */      0x6,                    /* FC_SHORT */
                                       0x1,                    /*
FC_BYTE */
/* 962 */      0x1,                    /* FC_BYTE */
                                       0x8,                    /*
FC_LONG */
/* 964 */      0xb,                    /* FC_HYPER */
                                       0x5b,                   /*
FC_END */
/* 966 */
                                       0x12, 0x0,    /* FC_UP */
/* 968 */      NdrFcShort( 0xfffffff4 ),  /* Offset= -12 (956) */
/* 970 */
                                       0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 972 */      0x2,                    /* FC_CHAR */
                                       0x5c,                   /*
FC_PAD */
/* 974 */
                                       0x1a,                   /*
FC_BOGUS_STRUCT */
                                       0x7,                    /* 7
*/
/* 976 */      NdrFcShort( 0x20 ),     /* 32 */
/* 978 */      NdrFcShort( 0x0 ),      /* 0 */
/* 980 */      NdrFcShort( 0x0 ),      /* Offset= 0 (980) */
/* 982 */      0x8,                    /* FC_LONG */
                                       0x8,                    /*
FC_LONG */
/* 984 */      0x6,                    /* FC_SHORT */
                                       0x6,                    /*
FC_SHORT */
/* 986 */      0x6,                    /* FC_SHORT */
                                       0x6,                    /*
FC_SHORT */
/* 988 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                                       0x0,                    /* 0
*/
/* 990 */      NdrFcShort( 0xfffffc28 ),  /* Offset= -984 (6) */
/* 992 */      0x5c,                   /* FC_PAD */
```

```
                                    0x5b,                   /*
FC_END */
/* 994 */      0xb4,                /* FC_USER_MARSHAL */
                                    0x83,                   /*
131 */
/* 996 */      NdrFcShort( 0x0 ),   /* 0 */
/* 998 */      NdrFcShort( 0x10 ),  /* 16 */
/* 1000 */     NdrFcShort( 0x0 ),   /* 0 */
/* 1002 */     NdrFcShort( 0xfffffc18 ),   /* Offset= -1000 (2) */
/* 1004 */
                                    0x11, 0x4,    /* FC_RP
[alloced_on_stack] */
/* 1006 */     NdrFcShort( 0x6 ),   /* Offset= 6 (1012) */
/* 1008 */
                                    0x13, 0x0,    /* FC_OP */
/* 1010 */     NdrFcShort( 0xfffffffdc ),  /* Offset= -36 (974) */
/* 1012 */     0xb4,                /* FC_USER_MARSHAL */
                                    0x83,                   /*
131 */
/* 1014 */     NdrFcShort( 0x0 ),   /* 0 */
/* 1016 */     NdrFcShort( 0x10 ),  /* 16 */
/* 1018 */     NdrFcShort( 0x0 ),   /* 0 */
/* 1020 */     NdrFcShort( 0xfffffff4 ),   /* Offset= -12 (1008) */

                                    0x0
        }
    };

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };



/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00}} */


/* Object interface: IUnknown, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,
0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    34,
    68,
    102,
    136,
    170
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };


static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */

};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0,   /* proxy/server info */
    0   /* Reserved5 */
    };

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};
```

```
#define _tpcc_com_ps_CHECK_IID(n)    IID_GENERIC_CHECK_IID(
_tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex
)
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};


#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/


#pragma warning( disable: 4049 )  /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */


 /* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win64 (32b run,appending)
    protocol : dce , ms_ext, c_ext, robust
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/

//@@MIDL_FILE_HEADING(  )
#if defined(_M_IA64) || defined(_M_AMD64)
#define USE_STUBLESS_PROXY


/* verify that the <rpcproxy.h> version is high enough to compile this
file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif


#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__


#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE   1003
#define PROC_FORMAT_STRING_SIZE   253
#define TRANSMIT_AS_TABLE_SIZE    0
#define WIRE_MARSHAL_TABLE_SIZE   1

typedef struct _MIDL_TYPE_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
    {
    short          Pad;
    unsigned char  Format[ PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;


static RPC_SYNTAX_IDENTIFIER  _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,
0x60}},{2,0}};


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

                /* Procedure NewOrder */

                                        0x33,                   /*
FC_AUTO_HANDLE */
                                        0x6c,                   /*
Old Flags:  object, Oi2 */
/*  2 */        NdrFcLong( 0x0 ),       /* 0 */
/*  6 */        NdrFcShort( 0x3 ),      /* 3 */
/*  8 */        NdrFcShort( 0x30 ),     /* ia64 Stack size/offset = 48
*/
/* 10 */        NdrFcShort( 0x0 ),      /* 0 */
/* 12 */        NdrFcShort( 0x8 ),      /* 8 */
/* 14 */        0x47,                   /* Oi2 Flags:  srv must size, clt
must size, has return, has ext, */
                                        0x3,                    /* 3
*/
/* 16 */        0xa,                    /* 10 */
                                        0x7,                    /*
Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 18 */        NdrFcShort( 0x20 ),     /* 32 */
/* 20 */        NdrFcShort( 0x20 ),     /* 32 */
/* 22 */        NdrFcShort( 0x0 ),      /* 0 */
/* 24 */        NdrFcShort( 0x0 ),      /* 0 */

                /* Parameter txn_in */

/* 26 */        NdrFcShort( 0x8b ),     /* Flags:  must size, must free,
in, by val, */
/* 28 */        NdrFcShort( 0x8 ),      /* ia64 Stack size/offset = 8 */
/* 30 */        NdrFcShort( 0x3ce ),    /* Type Offset=974 */

                /* Parameter txn_out */

/* 32 */        NdrFcShort( 0x6113 ),   /* Flags:  must size, must free,
out, simple ref, srv alloc size=24 */
/* 34 */        NdrFcShort( 0x20 ),     /* ia64 Stack size/offset = 32
*/
/* 36 */        NdrFcShort( 0x3e0 ),    /* Type Offset=992 */

                /* Return value */
```

```
/* 38 */      NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 40 */      NdrFcShort( 0x28 ),       /* ia64 Stack size/offset = 40
*/
/* 42 */      0x8,                      /* FC_LONG */
                                        0x0,                      /* 0
*/

            /* Procedure Payment */

/* 44 */      0x33,                     /* FC_AUTO_HANDLE */
                                        0x6c,                     /*
Old Flags:  object, Oi2 */
/* 46 */      NdrFcLong( 0x0 ),         /* 0 */
/* 50 */      NdrFcShort( 0x4 ),        /* 4 */
/* 52 */      NdrFcShort( 0x30 ),       /* ia64 Stack size/offset = 48
*/
/* 54 */      NdrFcShort( 0x0 ),        /* 0 */
/* 56 */      NdrFcShort( 0x8 ),        /* 8 */
/* 58 */      0x47,                     /* Oi2 Flags:  srv must size, clt
must size, has return, has ext, */
                                        0x3,                      /* 3
*/
/* 60 */      0xa,                      /* 10 */
                                        0x7,                      /*
Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 62 */      NdrFcShort( 0x20 ),       /* 32 */
/* 64 */      NdrFcShort( 0x20 ),       /* 32 */
/* 66 */      NdrFcShort( 0x0 ),        /* 0 */
/* 68 */      NdrFcShort( 0x0 ),        /* 0 */

            /* Parameter txn_in */

/* 70 */      NdrFcShort( 0x8b ),       /* Flags:  must size, must free,
in, by val, */
/* 72 */      NdrFcShort( 0x8 ),        /* ia64 Stack size/offset = 8 */
/* 74 */      NdrFcShort( 0x3ce ),      /* Type Offset=974 */

            /* Parameter txn_out */

/* 76 */      NdrFcShort( 0x6113 ),     /* Flags:  must size, must free,
out, simple ref, srv alloc size=24 */
/* 78 */      NdrFcShort( 0x20 ),       /* ia64 Stack size/offset = 32
*/
/* 80 */      NdrFcShort( 0x3e0 ),      /* Type Offset=992 */

            /* Return value */

/* 82 */      NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 84 */      NdrFcShort( 0x28 ),       /* ia64 Stack size/offset = 40
*/
/* 86 */      0x8,                      /* FC_LONG */
                                        0x0,                      /* 0
*/

            /* Procedure Delivery */

/* 88 */      0x33,                     /* FC_AUTO_HANDLE */
                                        0x6c,                     /*
Old Flags:  object, Oi2 */
/* 90 */      NdrFcLong( 0x0 ),         /* 0 */
/* 94 */      NdrFcShort( 0x5 ),        /* 5 */
/* 96 */      NdrFcShort( 0x30 ),       /* ia64 Stack size/offset = 48
*/
/* 98 */      NdrFcShort( 0x0 ),        /* 0 */
/* 100 */     NdrFcShort( 0x8 ),        /* 8 */
/* 102 */     0x47,                     /* Oi2 Flags:  srv must size, clt
must size, has return, has ext, */
                                        0x3,                      /* 3
*/
/* 104 */     0xa,                      /* 10 */
                                        0x7,                      /*
Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 106 */     NdrFcShort( 0x20 ),       /* 32 */
/* 108 */     NdrFcShort( 0x20 ),       /* 32 */
/* 110 */     NdrFcShort( 0x0 ),        /* 0 */
/* 112 */     NdrFcShort( 0x0 ),        /* 0 */

            /* Parameter txn_in */

/* 114 */     NdrFcShort( 0x8b ),       /* Flags:  must size, must free,
in, by val, */
/* 116 */     NdrFcShort( 0x8 ),        /* ia64 Stack size/offset = 8 */
/* 118 */     NdrFcShort( 0x3ce ),      /* Type Offset=974 */

            /* Parameter txn_out */

/* 120 */     NdrFcShort( 0x6113 ),     /* Flags:  must size, must free,
out, simple ref, srv alloc size=24 */
/* 122 */     NdrFcShort( 0x20 ),       /* ia64 Stack size/offset = 32
*/
/* 124 */     NdrFcShort( 0x3e0 ),      /* Type Offset=992 */

            /* Return value */

/* 126 */     NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 128 */     NdrFcShort( 0x28 ),       /* ia64 Stack size/offset = 40
*/
/* 130 */     0x8,                      /* FC_LONG */
                                        0x0,                      /* 0
*/

            /* Procedure StockLevel */

/* 132 */     0x33,                     /* FC_AUTO_HANDLE */
                                        0x6c,                     /*
Old Flags:  object, Oi2 */
/* 134 */     NdrFcLong( 0x0 ),         /* 0 */
/* 138 */     NdrFcShort( 0x6 ),        /* 6 */
/* 140 */     NdrFcShort( 0x30 ),       /* ia64 Stack size/offset = 48
*/
/* 142 */     NdrFcShort( 0x0 ),        /* 0 */
/* 144 */     NdrFcShort( 0x8 ),        /* 8 */
/* 146 */     0x47,                     /* Oi2 Flags:  srv must size, clt
must size, has return, has ext, */
                                        0x3,                      /* 3
*/
/* 148 */     0xa,                      /* 10 */
                                        0x7,                      /*
Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 150 */     NdrFcShort( 0x20 ),       /* 32 */
/* 152 */     NdrFcShort( 0x20 ),       /* 32 */
/* 154 */     NdrFcShort( 0x0 ),        /* 0 */
/* 156 */     NdrFcShort( 0x0 ),        /* 0 */

            /* Parameter txn_in */

/* 158 */     NdrFcShort( 0x8b ),       /* Flags:  must size, must free,
in, by val, */
/* 160 */     NdrFcShort( 0x8 ),        /* ia64 Stack size/offset = 8 */
/* 162 */     NdrFcShort( 0x3ce ),      /* Type Offset=974 */

            /* Parameter txn_out */

/* 164 */     NdrFcShort( 0x6113 ),     /* Flags:  must size, must free,
out, simple ref, srv alloc size=24 */
/* 166 */     NdrFcShort( 0x20 ),       /* ia64 Stack size/offset = 32
*/
/* 168 */     NdrFcShort( 0x3e0 ),      /* Type Offset=992 */

            /* Return value */

/* 170 */     NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 172 */     NdrFcShort( 0x28 ),       /* ia64 Stack size/offset = 40
*/
/* 174 */     0x8,                      /* FC_LONG */
                                        0x0,                      /* 0
*/

            /* Procedure OrderStatus */

/* 176 */     0x33,                     /* FC_AUTO_HANDLE */
                                        0x6c,                     /*
Old Flags:  object, Oi2 */
/* 178 */     NdrFcLong( 0x0 ),         /* 0 */
/* 182 */     NdrFcShort( 0x7 ),        /* 7 */
```

```
/* 184 */      NdrFcShort( 0x30 ),       /* ia64 Stack size/offset = 48
*/
/* 186 */      NdrFcShort( 0x0 ),        /* 0 */
/* 188 */      NdrFcShort( 0x8 ),        /* 8 */
/* 190 */      0x47,                     /* Oi2 Flags:  srv must size, clt
must size, has return, has ext, */
                                         0x3,                      /* 3
*/
/* 192 */      0xa,                      /* 10 */
                                         0x7,                      /*
Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 194 */      NdrFcShort( 0x20 ),       /* 32 */
/* 196 */      NdrFcShort( 0x20 ),       /* 32 */
/* 198 */      NdrFcShort( 0x0 ),        /* 0 */
/* 200 */      NdrFcShort( 0x0 ),        /* 0 */

               /* Parameter txn_in */

/* 202 */      NdrFcShort( 0x8b ),       /* Flags:  must size, must free,
in, by val, */
/* 204 */      NdrFcShort( 0x8 ),        /* ia64 Stack size/offset = 8 */
/* 206 */      NdrFcShort( 0x3ce ),      /* Type Offset=974 */

               /* Parameter txn_out */

/* 208 */      NdrFcShort( 0x6113 ),     /* Flags:  must size, must free,
out, simple ref, srv alloc size=24 */
/* 210 */      NdrFcShort( 0x20 ),       /* ia64 Stack size/offset = 32
*/
/* 212 */      NdrFcShort( 0x3e0 ),      /* Type Offset=992 */

               /* Return value */

/* 214 */      NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 216 */      NdrFcShort( 0x28 ),       /* ia64 Stack size/offset = 40
*/
/* 218 */      0x8,                      /* FC_LONG */
                                         0x0,                      /* 0
*/

               /* Procedure CallSetComplete */

/* 220 */      0x33,                     /* FC_AUTO_HANDLE */
                                         0x6c,                     /*
Old Flags:  object, Oi2 */
/* 222 */      NdrFcLong( 0x0 ),         /* 0 */
/* 226 */      NdrFcShort( 0x8 ),        /* 8 */
/* 228 */      NdrFcShort( 0x10 ),       /* ia64 Stack size/offset = 16
*/
/* 230 */      NdrFcShort( 0x0 ),        /* 0 */
/* 232 */      NdrFcShort( 0x8 ),        /* 8 */
/* 234 */      0x44,                     /* Oi2 Flags:  has return, has
ext, */
                                         0x1,                      /* 1
*/
/* 236 */      0xa,                      /* 10 */
                                         0x1,                      /*
Ext Flags:  new corr desc, */
/* 238 */      NdrFcShort( 0x0 ),        /* 0 */
/* 240 */      NdrFcShort( 0x0 ),        /* 0 */
/* 242 */      NdrFcShort( 0x0 ),        /* 0 */
/* 244 */      NdrFcShort( 0x0 ),        /* 0 */

               /* Return value */

/* 246 */      NdrFcShort( 0x70 ),       /* Flags:  out, return, base
type, */
/* 248 */      NdrFcShort( 0x8 ),        /* ia64 Stack size/offset = 8 */
/* 250 */      0x8,                      /* FC_LONG */
                                         0x0,                      /* 0
*/

                                         0x0
    }
  };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
  {
    0,
    {
                                         NdrFcShort( 0x0 ),        /* 0
*/
/*  2 */
                                         0x12, 0x0,    /* FC_UP */
/*  4 */       NdrFcShort( 0x3b6 ),      /* Offset= 950 (954) */
/*  6 */
                                         0x2b,                     /*
FC_NON_ENCAPSULATED_UNION */
                                         0x9,                      /*
FC_ULONG */
/*  8 */       0x7,                      /* Corr desc: FC_USHORT */
                                         0x0,                      /*
*/
/* 10 */       NdrFcShort( 0xfff8 ),     /* -8 */
/* 12 */       NdrFcShort( 0x1 ),        /* Corr flags:  early, */
/* 14 */       NdrFcShort( 0x2 ),        /* Offset= 2 (16) */
/* 16 */       NdrFcShort( 0x10 ),       /* 16 */
/* 18 */       NdrFcShort( 0x2f ),       /* 47 */
/* 20 */       NdrFcLong( 0x14 ),        /* 20 */
/* 24 */       NdrFcShort( 0x800b ),     /* Simple arm type: FC_HYPER
*/
/* 26 */       NdrFcLong( 0x3 ),         /* 3 */
/* 30 */       NdrFcShort( 0x8008 ),     /* Simple arm type: FC_LONG
*/
/* 32 */       NdrFcLong( 0x11 ),        /* 17 */
/* 36 */       NdrFcShort( 0x8001 ),     /* Simple arm type: FC_BYTE
*/
/* 38 */       NdrFcLong( 0x2 ),         /* 2 */
/* 42 */       NdrFcShort( 0x8006 ),     /* Simple arm type: FC_SHORT
*/
/* 44 */       NdrFcLong( 0x4 ),         /* 4 */
/* 48 */       NdrFcShort( 0x800a ),     /* Simple arm type: FC_FLOAT
*/
/* 50 */       NdrFcLong( 0x5 ),         /* 5 */
/* 54 */       NdrFcShort( 0x800c ),     /* Simple arm type:
FC_DOUBLE */
/* 56 */       NdrFcLong( 0xb ),         /* 11 */
/* 60 */       NdrFcShort( 0x8006 ),     /* Simple arm type: FC_SHORT
*/
/* 62 */       NdrFcLong( 0xa ),         /* 10 */
/* 66 */       NdrFcShort( 0x8008 ),     /* Simple arm type: FC_LONG
*/
/* 68 */       NdrFcLong( 0x6 ),         /* 6 */
/* 72 */       NdrFcShort( 0xe8 ),       /* Offset= 232 (304) */
/* 74 */       NdrFcLong( 0x7 ),         /* 7 */
/* 78 */       NdrFcShort( 0x800c ),     /* Simple arm type:
FC_DOUBLE */
/* 80 */       NdrFcLong( 0x8 ),         /* 8 */
/* 84 */       NdrFcShort( 0xe2 ),       /* Offset= 226 (310) */
/* 86 */       NdrFcLong( 0xd ),         /* 13 */
/* 90 */       NdrFcShort( 0xf6 ),       /* Offset= 246 (336) */
/* 92 */       NdrFcLong( 0x9 ),         /* 9 */
/* 96 */       NdrFcShort( 0x102 ),      /* Offset= 258 (354) */
/* 98 */       NdrFcLong( 0x2000 ),      /* 8192 */
/* 102 */      NdrFcShort( 0x10e ),      /* Offset= 270 (372) */
/* 104 */      NdrFcLong( 0x24 ),        /* 36 */
/* 108 */      NdrFcShort( 0x304 ),      /* Offset= 772 (880) */
/* 110 */      NdrFcLong( 0x4024 ),      /* 16420 */
/* 114 */      NdrFcShort( 0x2fe ),      /* Offset= 766 (880) */
/* 116 */      NdrFcLong( 0x4011 ),      /* 16401 */
/* 120 */      NdrFcShort( 0x2fc ),      /* Offset= 764 (884) */
/* 122 */      NdrFcLong( 0x4002 ),      /* 16386 */
/* 126 */      NdrFcShort( 0x2fa ),      /* Offset= 762 (888) */
/* 128 */      NdrFcLong( 0x4003 ),      /* 16387 */
/* 132 */      NdrFcShort( 0x2f8 ),      /* Offset= 760 (892) */
/* 134 */      NdrFcLong( 0x4014 ),      /* 16404 */
/* 138 */      NdrFcShort( 0x2f6 ),      /* Offset= 758 (896) */
/* 140 */      NdrFcLong( 0x4004 ),      /* 16388 */
/* 144 */      NdrFcShort( 0x2f4 ),      /* Offset= 756 (900) */
/* 146 */      NdrFcLong( 0x4005 ),      /* 16389 */
/* 150 */      NdrFcShort( 0x2f2 ),      /* Offset= 754 (904) */
/* 152 */      NdrFcLong( 0x400b ),      /* 16395 */
/* 156 */      NdrFcShort( 0x2dc ),      /* Offset= 732 (888) */
/* 158 */      NdrFcLong( 0x400a ),      /* 16394 */
/* 162 */      NdrFcShort( 0x2da ),      /* Offset= 730 (892) */
/* 164 */      NdrFcLong( 0x4006 ),      /* 16390 */
/* 168 */      NdrFcShort( 0x2e4 ),      /* Offset= 740 (908) */
/* 170 */      NdrFcLong( 0x4007 ),      /* 16391 */
/* 174 */      NdrFcShort( 0x2da ),      /* Offset= 730 (904) */
```

```
/* 176 */    NdrFcLong( 0x4008 ),      /* 16392 */
/* 180 */    NdrFcShort( 0x2dc ),      /* Offset= 732 (912) */
/* 182 */    NdrFcLong( 0x400d ),      /* 16397 */
/* 186 */    NdrFcShort( 0x2da ),      /* Offset= 730 (916) */
/* 188 */    NdrFcLong( 0x4009 ),      /* 16393 */
/* 192 */    NdrFcShort( 0x2d8 ),      /* Offset= 728 (920) */
/* 194 */    NdrFcLong( 0x6000 ),      /* 24576 */
/* 198 */    NdrFcShort( 0x2d6 ),      /* Offset= 726 (924) */
/* 200 */    NdrFcLong( 0x400c ),      /* 16396 */
/* 204 */    NdrFcShort( 0x2d4 ),      /* Offset= 724 (928) */
/* 206 */    NdrFcLong( 0x10 ),        /* 16 */
/* 210 */    NdrFcShort( 0x8002 ),     /* Simple arm type: FC_CHAR
*/
/* 212 */    NdrFcLong( 0x12 ),        /* 18 */
/* 216 */    NdrFcShort( 0x8006 ),     /* Simple arm type: FC_SHORT
*/
/* 218 */    NdrFcLong( 0x13 ),        /* 19 */
/* 222 */    NdrFcShort( 0x8008 ),     /* Simple arm type: FC_LONG
*/
/* 224 */    NdrFcLong( 0x15 ),        /* 21 */
/* 228 */    NdrFcShort( 0x800b ),     /* Simple arm type: FC_HYPER
*/
/* 230 */    NdrFcLong( 0x16 ),        /* 22 */
/* 234 */    NdrFcShort( 0x8008 ),     /* Simple arm type: FC_LONG
*/
/* 236 */    NdrFcLong( 0x17 ),        /* 23 */
/* 240 */    NdrFcShort( 0x8008 ),     /* Simple arm type: FC_LONG
*/
/* 242 */    NdrFcLong( 0xe ),         /* 14 */
/* 246 */    NdrFcShort( 0x2b2 ),      /* Offset= 690 (936) */
/* 248 */    NdrFcLong( 0x400e ),      /* 16398 */
/* 252 */    NdrFcShort( 0x2b6 ),      /* Offset= 694 (946) */
/* 254 */    NdrFcLong( 0x4010 ),      /* 16400 */
/* 258 */    NdrFcShort( 0x2b4 ),      /* Offset= 692 (950) */
/* 260 */    NdrFcLong( 0x4012 ),      /* 16402 */
/* 264 */    NdrFcShort( 0x270 ),      /* Offset= 624 (888) */
/* 266 */    NdrFcLong( 0x4013 ),      /* 16403 */
/* 270 */    NdrFcShort( 0x26e ),      /* Offset= 622 (892) */
/* 272 */    NdrFcLong( 0x4015 ),      /* 16405 */
/* 276 */    NdrFcShort( 0x26c ),      /* Offset= 620 (896) */
/* 278 */    NdrFcLong( 0x4016 ),      /* 16406 */
/* 282 */    NdrFcShort( 0x262 ),      /* Offset= 610 (892) */
/* 284 */    NdrFcLong( 0x4017 ),      /* 16407 */
/* 288 */    NdrFcShort( 0x25c ),      /* Offset= 604 (892) */
/* 290 */    NdrFcLong( 0x0 ),         /* 0 */
/* 294 */    NdrFcShort( 0x0 ),        /* Offset= 0 (294) */
/* 296 */    NdrFcLong( 0x1 ),         /* 1 */
/* 300 */    NdrFcShort( 0x0 ),        /* Offset= 0 (300) */
/* 302 */    NdrFcShort( 0xffffffff ), /* Offset= -1 (301) */
/* 304 */
                                       0x15,                    /*
FC_STRUCT */
                                       0x7,                     /* 7
*/
/* 306 */    NdrFcShort( 0x8 ),        /* 8 */
/* 308 */    0xb,                      /* FC_HYPER */
                                       0x5b,                    /*
FC_END */
/* 310 */
                                       0x12, 0x0,    /* FC_UP */
/* 312 */    NdrFcShort( 0xe ),        /* Offset= 14 (326) */
/* 314 */
                                       0x1b,                    /*
FC_CARRAY */
                                       0x1,                     /* 1
*/
/* 316 */    NdrFcShort( 0x2 ),        /* 2 */
/* 318 */    0x9,                      /* Corr desc: FC_ULONG */
                                       0x0,                     /*
*/
/* 320 */    NdrFcShort( 0xfffc ),     /* -4 */
/* 322 */    NdrFcShort( 0x1 ),        /* Corr flags:  early, */
/* 324 */    0x6,                      /* FC_SHORT */
                                       0x5b,                    /*
FC_END */
/* 326 */
                                       0x17,                    /*
FC_CSTRUCT */
                                       0x3,                     /* 3
*/
/* 328 */    NdrFcShort( 0x8 ),        /* 8 */
/* 330 */    NdrFcShort( 0xfffffff0 ), /* Offset= -16 (314) */
/* 332 */    0x8,                      /* FC_LONG */
                                       0x8,                     /*
FC_LONG */
/* 334 */    0x5c,                     /* FC_PAD */
                                       0x5b,                    /*
FC_END */
/* 336 */
                                       0x2f,                    /*
FC_IP */
                                       0x5a,                    /*
FC_CONSTANT_IID */
/* 338 */    NdrFcLong( 0x0 ),         /* 0 */
/* 342 */    NdrFcShort( 0x0 ),        /* 0 */
/* 344 */    NdrFcShort( 0x0 ),        /* 0 */
/* 346 */    0xc0,                     /* 192 */
                                       0x0,                     /* 0
*/
/* 348 */    0x0,                      /* 0 */
                                       0x0,                     /* 0
*/
/* 350 */    0x0,                      /* 0 */
                                       0x0,                     /* 0
*/
/* 352 */    0x0,                      /* 0 */
                                       0x46,                    /*
70 */
/* 354 */
                                       0x2f,                    /*
FC_IP */
                                       0x5a,                    /*
FC_CONSTANT_IID */
/* 356 */    NdrFcLong( 0x20400 ),     /* 132096 */
/* 360 */    NdrFcShort( 0x0 ),        /* 0 */
/* 362 */    NdrFcShort( 0x0 ),        /* 0 */
/* 364 */    0xc0,                     /* 192 */
                                       0x0,                     /* 0
*/
/* 366 */    0x0,                      /* 0 */
                                       0x0,                     /* 0
*/
/* 368 */    0x0,                      /* 0 */
                                       0x0,                     /* 0
*/
/* 370 */    0x0,                      /* 0 */
                                       0x46,                    /*
70 */
/* 372 */
                                       0x12, 0x10,  /* FC_UP
[pointer_deref] */
/* 374 */    NdrFcShort( 0x2 ),        /* Offset= 2 (376) */
/* 376 */
                                       0x12, 0x0,    /* FC_UP */
/* 378 */    NdrFcShort( 0x1e4 ),      /* Offset= 484 (862) */
/* 380 */
                                       0x2a,                    /*
FC_ENCAPSULATED_UNION */
                                       0x89,                    /*
137 */
/* 382 */    NdrFcShort( 0x20 ),       /* 32 */
/* 384 */    NdrFcShort( 0xa ),        /* 10 */
/* 386 */    NdrFcLong( 0x8 ),         /* 8 */
/* 390 */    NdrFcShort( 0x50 ),       /* Offset= 80 (470) */
/* 392 */    NdrFcLong( 0xd ),         /* 13 */
/* 396 */    NdrFcShort( 0x70 ),       /* Offset= 112 (508) */
/* 398 */    NdrFcLong( 0x9 ),         /* 9 */
/* 402 */    NdrFcShort( 0x90 ),       /* Offset= 144 (546) */
/* 404 */    NdrFcLong( 0xc ),         /* 12 */
/* 408 */    NdrFcShort( 0xb0 ),       /* Offset= 176 (584) */
/* 410 */    NdrFcLong( 0x24 ),        /* 36 */
/* 414 */    NdrFcShort( 0x102 ),      /* Offset= 258 (672) */
/* 416 */    NdrFcLong( 0x800d ),      /* 32781 */
/* 420 */    NdrFcShort( 0x11e ),      /* Offset= 286 (706) */
/* 422 */    NdrFcLong( 0x10 ),        /* 16 */
/* 426 */    NdrFcShort( 0x138 ),      /* Offset= 312 (738) */
/* 428 */    NdrFcLong( 0x2 ),         /* 2 */
/* 432 */    NdrFcShort( 0x14e ),      /* Offset= 334 (766) */
/* 434 */    NdrFcLong( 0x3 ),         /* 3 */
/* 438 */    NdrFcShort( 0x164 ),      /* Offset= 356 (794) */
```

```
/* 440 */    NdrFcLong( 0x14 ),          /* 20 */
/* 444 */    NdrFcShort( 0x17a ),        /* Offset= 378 (822) */
/* 446 */    NdrFcShort( 0xffffffff ),   /* Offset= -1 (445) */
/* 448 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3
*/
/* 450 */    NdrFcShort( 0x0 ),          /* 0 */
/* 452 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 454 */    NdrFcShort( 0x0 ),          /* 0 */
/* 456 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 458 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 462 */    NdrFcShort( 0x0 ),          /* Corr flags:  */
/* 464 */
                                         0x12, 0x0,    /* FC_UP */
/* 466 */    NdrFcShort( 0xffffff74 ),   /* Offset= -140 (326) */
/* 468 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /*
FC_END */
/* 470 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 472 */    NdrFcShort( 0x10 ),         /* 16 */
/* 474 */    NdrFcShort( 0x0 ),          /* 0 */
/* 476 */    NdrFcShort( 0x6 ),          /* Offset= 6 (482) */
/* 478 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 480 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 482 */
                                         0x11, 0x0,    /* FC_RP */
/* 484 */    NdrFcShort( 0xffffffdc ),   /* Offset= -36 (448) */
/* 486 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3
*/
/* 488 */    NdrFcShort( 0x0 ),          /* 0 */
/* 490 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 492 */    NdrFcShort( 0x0 ),          /* 0 */
/* 494 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 496 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 500 */    NdrFcShort( 0x0 ),          /* Corr flags:  */
/* 502 */    0x4c,                       /* FC_EMBEDDED_COMPLEX
*/
                                         0x0,                        /* 0
*/
/* 504 */    NdrFcShort( 0xffffff58 ),   /* Offset= -168 (336) */
/* 506 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /*
FC_END */
/* 508 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 510 */    NdrFcShort( 0x10 ),         /* 16 */
/* 512 */    NdrFcShort( 0x0 ),          /* 0 */
/* 514 */    NdrFcShort( 0x6 ),          /* Offset= 6 (520) */
/* 516 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 518 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 520 */
                                         0x11, 0x0,    /* FC_RP */
/* 522 */    NdrFcShort( 0xffffffdc ),   /* Offset= -36 (486) */
/* 524 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3
*/
/* 526 */    NdrFcShort( 0x0 ),          /* 0 */
/* 528 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 530 */    NdrFcShort( 0x0 ),          /* 0 */
/* 532 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 534 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 538 */    NdrFcShort( 0x0 ),          /* Corr flags:  */
/* 540 */    0x4c,                       /* FC_EMBEDDED_COMPLEX
*/
                                         0x0,                        /* 0
*/
/* 542 */    NdrFcShort( 0xffffff44 ),   /* Offset= -188 (354) */
/* 544 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /*
FC_END */
/* 546 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 548 */    NdrFcShort( 0x0 ),          /* 0 */
/* 550 */    NdrFcShort( 0x6 ),          /* Offset= 6 (556) */
/* 552 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 554 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 556 */
                                         0x11, 0x0,    /* FC_RP */
/* 558 */    NdrFcShort( 0xffffffdc ),   /* Offset= -36 (522) */
/* 560 */
                                         0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3
*/
/* 562 */    NdrFcShort( 0x0 ),          /* 0 */
/* 564 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 566 */    NdrFcShort( 0x0 ),          /* 0 */
/* 568 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 570 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 572 */    NdrFcShort( 0x0 ),          /* Corr flags:  */
/* 576 */
                                         0x12, 0x0,    /* FC_UP */
/* 578 */    NdrFcShort( 0x176 ),        /* Offset= 374 (952) */
/* 580 */    0x5c,                       /* FC_PAD */
/* 582 */
                                         0x5b,                       /*
FC_END */
/* 584 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 586 */    NdrFcShort( 0x10 ),         /* 16 */
/* 588 */    NdrFcShort( 0x0 ),          /* 0 */
/* 590 */    NdrFcShort( 0x6 ),          /* Offset= 6 (596) */
/* 592 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 594 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 596 */
                                         0x11, 0x0,    /* FC_RP */
/* 598 */    NdrFcShort( 0xffffffdc ),   /* Offset= -36 (562) */
/* 600 */
                                         0x2f,                       /*
FC_IP */
                                         0x5a,                       /*
FC_CONSTANT_IID */
```

```
/* 602 */    NdrFcLong( 0x2f ),          /* 47 */
/* 606 */    NdrFcShort( 0x0 ),          /* 0 */
/* 608 */    NdrFcShort( 0x0 ),          /* 0 */
/* 610 */    0xc0,                       /* 192 */
                                         0x0,                        /* 0
*/
/* 612 */    0x0,                        /* 0 */
                                         0x0,                        /* 0
*/
/* 614 */    0x0,                        /* 0 */
                                         0x0,                        /* 0
*/
/* 616 */    0x0,                        /* 0 */
                                         0x46,                       /*
70 */
/* 618 */
                                         0x1b,                       /*
FC_CARRAY */
                                         0x0,                        /* 0
*/
/* 620 */    NdrFcShort( 0x1 ),          /* 1 */
/* 622 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 624 */    NdrFcShort( 0x4 ),          /* 4 */
/* 626 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 628 */    0x1,                        /* FC_BYTE */
                                         0x5b,                       /*
FC_END */
/* 630 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 632 */    NdrFcShort( 0x18 ),         /* 24 */
/* 634 */    NdrFcShort( 0x0 ),          /* 0 */
/* 636 */    NdrFcShort( 0xa ),          /* Offset= 10 (646) */
/* 638 */    0x8,                        /* FC_LONG */
                                         0x8,                        /*
FC_LONG */
/* 640 */    0x4c,                       /* FC_EMBEDDED_COMPLEX
*/
                                         0x0,                        /* 0
*/
/* 642 */    NdrFcShort( 0xffffffd6 ),   /* Offset= -42 (600) */
/* 644 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 646 */
                                         0x12, 0x0,    /* FC_UP */
/* 648 */    NdrFcShort( 0xfffffffe2 ),  /* Offset= -30 (618) */
```

```
/* 650 */                               0x21,                       /*
FC_BOGUS_ARRAY */
                                         0x3,                        /* 3
*/
/* 652 */    NdrFcShort( 0x0 ),          /* 0 */
/* 654 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 656 */    NdrFcShort( 0x0 ),          /* 0 */
/* 658 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 660 */    NdrFcLong( 0xffffffff ),    /* -1 */
/* 664 */    NdrFcShort( 0x0 ),          /* Corr flags:  */
/* 666 */
                                         0x12, 0x0,    /* FC_UP */
/* 668 */    NdrFcShort( 0xfffffffda ),  /* Offset= -38 (630) */
/* 670 */    0x5c,                       /* FC_PAD */
                                         0x5b,                       /*
FC_END */
/* 672 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 674 */    NdrFcShort( 0x10 ),         /* 16 */
/* 676 */    NdrFcShort( 0x0 ),          /* 0 */
/* 678 */    NdrFcShort( 0x6 ),          /* Offset= 6 (684) */
/* 680 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 682 */    0x36,                       /* FC_POINTER */
                                         0x5b,                       /*
FC_END */
/* 684 */
                                         0x11, 0x0,    /* FC_RP */
/* 686 */    NdrFcShort( 0xfffffffdc ),  /* Offset= -36 (650) */
/* 688 */
                                         0x1d,                       /*
FC_SMFARRAY */
                                         0x0,                        /* 0
*/
/* 690 */    NdrFcShort( 0x8 ),          /* 8 */
/* 692 */    0x1,                        /* FC_BYTE */
                                         0x5b,                       /*
FC_END */
/* 694 */
                                         0x15,                       /*
FC_STRUCT */
                                         0x3,                        /* 3
*/
/* 696 */    NdrFcShort( 0x10 ),         /* 16 */
/* 698 */    0x8,                        /* FC_LONG */
```

```
                                         0x6,                        /*
/* 700 */                               0x6,                        /*
FC_SHORT */
                                         0x4c,                       /*
FC_EMBEDDED_COMPLEX */
/* 702 */    0x0,                        /* 0 */
                                         NdrFcShort( 0xfffffff1 ),   /*
Offset= -15 (688) */
                                         0x5b,                       /*
FC_END */
/* 706 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 708 */    NdrFcShort( 0x20 ),         /* 32 */
/* 710 */    NdrFcShort( 0x0 ),          /* 0 */
/* 712 */    NdrFcShort( 0xa ),          /* Offset= 10 (722) */
/* 714 */    0x8,                        /* FC_LONG */
                                         0x40,                       /*
FC_STRUCTPAD4 */
/* 716 */    0x36,                       /* FC_POINTER */
                                         0x4c,                       /*
FC_EMBEDDED_COMPLEX */
/* 718 */    0x0,                        /* 0 */
                                         NdrFcShort( 0xfffffffe7 ),  /*
Offset= -25 (694) */
                                         0x5b,                       /*
FC_END */
/* 722 */
                                         0x11, 0x0,    /* FC_RP */
/* 724 */    NdrFcShort( 0xffffff12 ),   /* Offset= -238 (486) */
/* 726 */
                                         0x1b,                       /*
FC_CARRAY */
                                         0x0,                        /* 0
*/
/* 728 */    NdrFcShort( 0x1 ),          /* 1 */
/* 730 */    0x19,                       /* Corr desc:  field pointer,
FC_ULONG */
                                         0x0,                        /*
*/
/* 732 */    NdrFcShort( 0x0 ),          /* 0 */
/* 734 */    NdrFcShort( 0x1 ),          /* Corr flags:  early, */
/* 736 */    0x1,                        /* FC_BYTE */
                                         0x5b,                       /*
FC_END */
/* 738 */
                                         0x1a,                       /*
FC_BOGUS_STRUCT */
                                         0x3,                        /* 3
*/
/* 740 */    NdrFcShort( 0x10 ),         /* 16 */
/* 742 */    NdrFcShort( 0x0 ),          /* 0 */
```

```
/* 744 */      NdrFcShort( 0x6 ),
/* 746 */      0x8,

FC_STRUCTPAD4 */
/* 748 */      0x36,

FC_END */
/* 750 */

/* 752 */      NdrFcShort( 0xffffffe6 ),
/* 754 */

FC_CARRAY */

*/
/* 756 */      NdrFcShort( 0x2 ),
/* 758 */      0x19,
FC_ULONG */

*/
/* 760 */      NdrFcShort( 0x0 ),
/* 762 */      NdrFcShort( 0x1 ),
/* 764 */      0x6,

FC_END */
/* 766 */

FC_BOGUS_STRUCT */

*/
/* 768 */      NdrFcShort( 0x10 ),
/* 770 */      NdrFcShort( 0x0 ),
/* 772 */      NdrFcShort( 0x6 ),
/* 774 */      0x8,

FC_STRUCTPAD4 */
/* 776 */      0x36,

FC_END */
/* 778 */

/* 780 */      NdrFcShort( 0xffffffe6 ),
/* 782 */

FC_CARRAY */

*/
/* 784 */      NdrFcShort( 0x4 ),
/* 786 */      0x19,
FC_ULONG */

*/
/* Offset= 6 (750) */
/* FC_LONG */
0x40,                   /*

/* FC_POINTER */
0x5b,                   /*


0x12, 0x0,    /* FC_UP */
/* Offset= -26 (726) */

0x1b,                   /*

0x1,                    /* 1

/* 2 */
/* Corr desc:  field pointer,

0x0,                    /*

/* 0 */
/* Corr flags:  early, */
/* FC_SHORT */
0x5b,                   /*


0x1a,                   /*

0x3,                    /* 3

/* 16 */
/* 0 */
/* Offset= 6 (778) */
/* FC_LONG */
0x40,                   /*

/* FC_POINTER */
0x5b,                   /*


0x12, 0x0,    /* FC_UP */
/* Offset= -26 (754) */

0x1b,                   /*

0x3,                    /* 3

/* 4 */
/* Corr desc:  field pointer,

0x0,                    /*
```

```
/* 788 */      NdrFcShort( 0x0 ),
/* 790 */      NdrFcShort( 0x0 ),
/* 792 */      0x8,

FC_END */
/* 794 */

FC_BOGUS_STRUCT */

*/
/* 796 */      NdrFcShort( 0x10 ),
/* 798 */      NdrFcShort( 0x0 ),
/* 800 */      NdrFcShort( 0x6 ),
/* 802 */      0x8,

FC_STRUCTPAD4 */
/* 804 */      0x36,

FC_END */
/* 806 */

/* 808 */      NdrFcShort( 0xffffffe6 ),
/* 810 */

FC_CARRAY */

*/
/* 812 */      NdrFcShort( 0x8 ),
/* 814 */      0x19,
FC_ULONG */

*/
/* 816 */      NdrFcShort( 0x0 ),
/* 818 */      NdrFcShort( 0x1 ),
/* 820 */      0xb,

FC_END */
/* 822 */

FC_BOGUS_STRUCT */

*/
/* 824 */      NdrFcShort( 0x10 ),
/* 826 */      NdrFcShort( 0x0 ),
/* 828 */      NdrFcShort( 0x6 ),
/* 830 */      0x8,

FC_STRUCTPAD4 */
/* 832 */      0x36,

FC_END */
/* 834 */

/* 836 */      NdrFcShort( 0xffffffe6 ),
/* Corr flags:  early, */
/* FC_LONG */
0x5b,                   /*


0x1a,                   /*

0x3,                    /* 3

/* 16 */
/* 0 */
/* Offset= 6 (806) */
/* FC_LONG */
0x40,                   /*

/* FC_POINTER */
0x5b,                   /*


0x12, 0x0,    /* FC_UP */
/* Offset= -26 (782) */

0x1b,                   /*

0x7,                    /* 7

/* 8 */
/* Corr desc:  field pointer,

0x0,                    /*

/* 0 */
/* Corr flags:  early, */
/* FC_HYPER */
0x5b,                   /*


0x1a,                   /*

0x3,                    /* 3

/* 16 */
/* 0 */
/* Offset= 6 (834) */
/* FC_LONG */
0x40,                   /*

/* FC_POINTER */
0x5b,                   /*


0x12, 0x0,    /* FC_UP */
/* Offset= -26 (810) */
```

```
/* 838 */                   0x15,                   /*
FC_STRUCT */
                            0x3,                    /* 3
*/
/* 840 */      NdrFcShort( 0x8 ),      /* 8 */
/* 842 */      0x8,                    /* FC_LONG */
                            0x8,                    /*
FC_LONG */
/* 844 */      0x5c,                   /* FC_PAD */
                            0x5b,                   /*
FC_END */
/* 846 */

                            0x1b,                   /*
FC_CARRAY */
                            0x3,                    /* 3
*/
/* 848 */      NdrFcShort( 0x8 ),      /* 8 */
/* 850 */      0x7,                    /* Corr desc: FC_USHORT */
                            0x0,                    /*
*/
/* 852 */      NdrFcShort( 0xffc8 ),   /* -56 */
/* 854 */      NdrFcShort( 0x1 ),      /* Corr flags:  early, */
/* 856 */      0x4c,                   /* FC_EMBEDDED_COMPLEX
*/
                            0x0,                    /* 0
*/
/* 858 */      NdrFcShort( 0xffffffec ),   /* Offset= -20 (838) */
/* 860 */      0x5c,                   /* FC_PAD */
                            0x5b,                   /*
FC_END */
/* 862 */

                            0x1a,                   /*
FC_BOGUS_STRUCT */
                            0x3,                    /* 3
*/
/* 864 */      NdrFcShort( 0x38 ),     /* 56 */
/* 866 */      NdrFcShort( 0xffffffec ),   /* Offset= -20 (846) */
/* 868 */      NdrFcShort( 0x0 ),      /* Offset= 0 (868) */
/* 870 */      0x6,                    /* FC_SHORT */
                            0x6,                    /*
FC_SHORT */
/* 872 */      0x8,                    /* FC_LONG */
                            0x8,                    /*
FC_LONG */
/* 874 */      0x40,                   /* FC_STRUCTPAD4 */
                            0x4c,                   /*
FC_EMBEDDED_COMPLEX */
/* 876 */      0x0,                    /* 0 */
                            NdrFcShort( 0xfffffe0f ),   /*
Offset= -497 (380) */
                            0x5b,                   /*
FC_END */
/* 880 */
```

```
                                0x12, 0x0,    /* FC_UP */
/* 882 */      NdrFcShort( 0xffffff04 ),   /* Offset= -252 (630) */
/* 884 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 886 */      0x1,                        /* FC_BYTE */
                                0x5c,                       /*
FC_PAD */
/* 888 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 890 */      0x6,                        /* FC_SHORT */
                                0x5c,                       /*
FC_PAD */
/* 892 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 894 */      0x8,                        /* FC_LONG */
                                0x5c,                       /*
FC_PAD */
/* 896 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 898 */      0xb,                        /* FC_HYPER */
                                0x5c,                       /*
FC_PAD */
/* 900 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 902 */      0xa,                        /* FC_FLOAT */
                                0x5c,                       /*
FC_PAD */
/* 904 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 906 */      0xc,                        /* FC_DOUBLE */
                                0x5c,                       /*
FC_PAD */
/* 908 */
                                0x12, 0x0,    /* FC_UP */
/* 910 */      NdrFcShort( 0xfffffda2 ),   /* Offset= -606 (304) */
/* 912 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 914 */      NdrFcShort( 0xfffffda4 ),   /* Offset= -604 (310) */
/* 916 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 918 */      NdrFcShort( 0xfffffdba ),   /* Offset= -582 (336) */
/* 920 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 922 */      NdrFcShort( 0xfffffdc8 ),   /* Offset= -568 (354) */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 926 */      NdrFcShort( 0xfffffdd6 ),   /* Offset= -554 (372) */
/* 928 */
                                0x12, 0x10,   /* FC_UP
[pointer_deref] */
/* 930 */      NdrFcShort( 0x2 ),          /* Offset= 2 (932) */
/* 932 */
                                0x12, 0x0,    /* FC_UP */
/* 934 */      NdrFcShort( 0x14 ),         /* Offset= 20 (954) */
/* 936 */
                                0x15,                       /*
FC_STRUCT */
                                0x7,                        /* 7
*/
/* 938 */      NdrFcShort( 0x10 ),         /* 16 */
/* 940 */      0x6,                        /* FC_SHORT */
                                0x1,                        /*
FC_BYTE */
/* 942 */      0x1,                        /* FC_BYTE */
                                0x8,                        /*
FC_LONG */
/* 944 */      0xb,                        /* FC_HYPER */
                                0x5b,                       /*
FC_END */
/* 946 */
                                0x12, 0x0,    /* FC_UP */
/* 948 */      NdrFcShort( 0xffffffff4 ),  /* Offset= -12 (936) */
/* 950 */
                                0x12, 0x8,    /* FC_UP
[simple_pointer] */
/* 952 */      0x2,                        /* FC_CHAR */
                                0x5c,                       /*
FC_PAD */
/* 954 */
                                0x1a,                       /*
FC_BOGUS_STRUCT */
                                0x7,                        /* 7
*/
/* 956 */      NdrFcShort( 0x20 ),         /* 32 */
/* 958 */      NdrFcShort( 0x0 ),          /* 0 */
/* 960 */      NdrFcShort( 0x0 ),          /* Offset= 0 (960) */
/* 962 */      0x8,                        /* FC_LONG */
                                0x8,                        /*
FC_LONG */
/* 964 */      0x6,                        /* FC_SHORT */
                                0x6,                        /*
FC_SHORT */
/* 966 */      0x6,                        /* FC_SHORT */
                                0x6,                        /*
FC_SHORT */
/* 968 */      0x4c,                       /* FC_EMBEDDED_COMPLEX
*/                              0x0,                        /* 0
*/
/* 970 */      NdrFcShort( 0xfffffc3c ),   /* Offset= -964 (6) */
/* 972 */      0x5c,                       /* FC_PAD */
                                0x5b,                       /*
FC_END */
/* 974 */      0xb4,                       /* FC_USER_MARSHAL */
                                0x83,                       /*
131 */
/* 976 */      NdrFcShort( 0x0 ),          /* 0 */
/* 978 */      NdrFcShort( 0x18 ),         /* 24 */
/* 980 */      NdrFcShort( 0x0 ),          /* 0 */
/* 982 */      NdrFcShort( 0xfffffc2c ),   /* Offset= -980 (2) */
/* 984 */
                                0x11, 0x4,    /* FC_RP
[alloced_on_stack] */
/* 986 */      NdrFcShort( 0x6 ),          /* Offset= 6 (992) */
/* 988 */
                                0x13, 0x0,    /* FC_OP */
/* 990 */      NdrFcShort( 0xfffffffdc ),  /* Offset= -36 (954) */
/* 992 */      0xb4,                       /* FC_USER_MARSHAL */
                                0x83,                       /*
131 */
/* 994 */      NdrFcShort( 0x0 ),          /* 0 */
/* 996 */      NdrFcShort( 0x18 ),         /* 24 */
/* 998 */      NdrFcShort( 0x0 ),          /* 0 */
/* 1000 */     NdrFcShort( 0xffffffff4 ),  /* Offset= -12 (988) */

                                0x0
    }
  };

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
    {

        {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
        }

    };



/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00}} */
```

```
/* Object interface: IUnknown, ver. 0.0,

GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x46}} */


/* Object interface: ITPCC, ver. 0.0,

GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,
0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    44,
    88,
    132,
    176,
    220
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
    };


static const MIDL_SERVER_INFO ITPCC_ServerInfo =
    {
    &Object_StubDesc,
    0,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0,   /* proxy/server info */
    0   /* Reserved5 */
    };

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    0ITPCC",
};


#define _tpcc_com_ps_CHECK_IID(n)    IID_GENERIC_CHECK_IID(
_tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex
)
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) & _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0  /* Filler3 */
};


#endif /* defined(_M_IA64) || defined(_M_AMD64)*/
```

Appendix B Database Load

Setup.cmd

```
@echo off

set DoDBCREATE=FALSE
set DoBULKINSERT=TRUE
set DoCLEANUP=TRUE
set DoBACKUP=FALSE

rem
rem Modify the following parameters for your configuration
rem

set HOMEDRIVE=G:
set HOMEDIR=\hptpcc
set OUTPUTDRIVE=G:
set OUTPUTDIR=\hptpcc\Output
set THREADS=16
set WAREHOUSES=112000
set DB=tpcc
set PASSWD=ssdl
set SERVER=tcp:sqlorca
set StartPortNo=1436

set HOMEPATH=%HOMEDRIVE%%HOMEDIR%
set FILEPATH=%FILEDRIVE%%FILEDIR%


if %PROCESSOR_ARCHITECTURE% == x86 (set
EXECUTABLEPATH=%HOMEPATH%\Executables\x86
) else set EXECUTABLEPATH=%HOMEPATH%\Executables\ipf

%HOMEDRIVE%
cd %HOMEDIR%

if '%1' == 'BULKINSERTn' goto :BULKINSERTn
if '%1' == 'DBCREATE' goto :DBCREATE

echo Checking for existence of HOMEDIR and OUTPUTDIR
if NOT EXIST %HOMEDRIVE%%HOMEDIR% goto :NO_HOMEDIR
if NOT EXIST %OUTPUTDRIVE%%OUTPUTDIR% goto :NO_OUTPUTDIR

echo Finding next output directory in %OUTPUTDRIVE%%OUTPUTDIR%
set OUTPUTNUMBER=1
:OUTPUTLOOP
  if NOT EXIST %OUTPUTDRIVE%%OUTPUTDIR%\%OUTPUTNUMBER%
goto :OUTPUTLOOPEND
  set /a OUTPUTNUMBER=%OUTPUTNUMBER%+1
  goto :OUTPUTLOOP
:OUTPUTLOOPEND
set OUTPUTPATH=%OUTPUTDRIVE%%OUTPUTDIR%\%OUTPUTNUMBER%
echo Output will be found at %OUTPUTPATH%
mkdir %OUTPUTPATH%

:DBCREATE
if NOT '%DoDBCREATE%' == 'TRUE' goto :BULKINSERT
rem
rem DBCREATE invokes the file CreateDatabase.sql
rem
echo Starting database creation
time /t >> %OUTPUTPATH%\time.txt
echo Starting database creation >> %OUTPUTPATH%\time.txt
%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E  -i CreateDatabase.sql -e -o %OUTPUTPATH%\CreateDatabase.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT
if '%1' == 'DBCREATE' %EXECUTABLEPATH%\Semaphore -release
echo DBCREATE done %OUTPUTPATH%\time.txt
time /t %OUTPUTPATH%\time.txt
if '%1' == 'DBCREATE' goto :EOF

:BULKINSERT
if NOT '%DoBULKINSERT%' == 'TRUE' goto :CLEANUP
rem
rem BULKINSERT loads each table through a set of concurrent bulk inserts
rem After each table is loaded, the table is indexed
rem

echo Starting Bulk Inserts
%HOMEDRIVE%
cd %HOMEDIR%\TablesAndIndexes

osql -E  -d%DB% -i%HOMEPATH%\Database\dbopt1.sql -e -o
%OUTPUTPATH%\dbopt1.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'show advanced options',1" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'max degree',32" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT

echo Dropping and Re-Creating Tables
osql -E -d%DB% -iCreateTables.sql -e -o %OUTPUTPATH%\CreateTables.out
-b
if ERRORLEVEL 1 goto :ERROR_EXIT
rem Special case for Item, since it is only one file

echo Starting create indexes for Item
osql -E  -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreateITEMIndexes.sql -e -
o%OUTPUTPATH%\NewCreateitemIndexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
echo Starting bulk inserts for Item
%EXECUTABLEPATH%\stockldr\tpccldr -s1 -W10 -titem -Pssdl -i1 -o1 -
f%OUTPUTPATH%\loaditem.out -L%OUTPUTPATH%\
if ERRORLEVEL 1 goto :ERROR_EXIT


echo Starting bulk inserts of item warehouse customer orders

for %%i in (warehouse customer orders) do call :BULKINSERTi %%i
if ERRORLEVEL 1 goto :ERROR_EXIT

goto :CLEANUP

:BULKINSERTi
echo Starting bulk inserts for %1
for /l %%j in (1,1,%THREADS%) do start cmd /C
%HOMEPATH%\SetupOldLoader BULKINSERTn %1 %%j
%EXECUTABLEPATH%\Semaphore -wait %THREADS%

echo Starting create indexes for %1
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreate%1Indexes.sql -e -o
%OUTPUTPATH%\NewCreate%1Indexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
goto :EOF


:BULKINSERTn
set /a REM=WAREHOUSES %% THREADS
set /a WAREHOUSEPERTHREAD=WAREHOUSES / THREADS
set /a TEMP=%3-1
set /a PortNo=%StartPortNo%+%3-1
set /a STARTWAREHOUSE=%TEMP%*%WAREHOUSEPERTHREAD%
if %3 leq %REM% set /a
WAREHOUSEPERTHREAD=%WAREHOUSEPERTHREAD%+1
if %3 leq %REM% set /a
STARTWAREHOUSE=%STARTWAREHOUSE%+%3%
if %3 gtr %REM% set /a
STARTWAREHOUSE=%STARTWAREHOUSE%+%REM%+1
set /a
ENDWAREHOUSE=%STARTWAREHOUSE%+%WAREHOUSEPERTHREAD%-1
```

```
%EXECUTABLEPATH%\stockldr\tpccldr -t%2 -Pssdl -
s%STARTWAREHOUSE% -W%ENDWAREHOUSE% -i1 -o1 -
f%OUTPUTPATH%\load%2%STARTWAREHOUSE%.out -L%OUTPUTPATH%\
-S%SERVER%,%PortNo%

%EXECUTABLEPATH%\Semaphore -release
goto :EOF


:CLEANUP
if NOT '%DoCLEANUP%' == 'TRUE' goto :BACKUP
rem
rem CLEANUP sets statistics and lock options
rem

echo Setting Cleanup Options

%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E -d%DB% -i%HOMEPATH%\Database\dbopt2.sql -e -o
%OUTPUTPATH%\dbopt2.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'max degree',1" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT


cd %HOMEDIR%\StoredProcedures
for %%i in (Create*Proc.sql) do osql -E -d%DB% -
i%HOMEPATH%\StoredProcedures\%%i -e -o %OUTPUTPATH%\%%i.out -b

osql -E -Q"sp_dboption %DB%,'trunc',FALSE" -b
>%OUTPUTPATH%\dboptionTrunOFF.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_dboption %DB%,'select',FALSE" -b
>%OUTPUTPATH%\dboptionSelOFF.out
if ERRORLEVEL 1 goto :ERROR_EXIT

:BACKUP
rem
rem BACKUP is the final step, using a script
rem

if NOT '%DoBACKUP%' == 'TRUE' goto :DONE
%HOMEDRIVE%
cd %HOMEDIR%\Database

echo Starting Backup
osql -E -d%DB% -i%HOMEPATH%\database\Backup.sql -e -o
%OUTPUTPATH%\Backup.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT

:DONE

echo Done!  Check for output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%

goto :EOF

:NO_HOMEDIR
echo directory %HOMEDRIVE%%HOMEDIR% does not exist
goto :ERROR_EXIT

:NO_OUTPUTDIR
echo directory %OUtPUTDRIVE%%OUTPUTDIR% does not exist
goto :ERROR_EXIT

:ERROR_EXIT
echo SetupOldLoader aborted due to errors
echo Check output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%
exit /B
```

tpcc.h

```
//=========================================
==============================
//          File:                    TPCCLDR.C
//                                   Microsoft TPC-C Kit
Ver. 4.51
//                                   Copyright Microsoft,
1996, 1997, 1998, 1999,
//                                   2000, 2001, 2002,
2003
//          Purpose:    Source file for TPC-C database loader
//=========================================
==============================
// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS            100000
#define MAXITEMS_SCALE_DOWN            100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN           30
#define ORDERS_PER_DISTRICT     3000
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_SCALE_DOWN      30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS      3
#define MAX_MAIN_THREADS       4
#define MAX_SQL_ERRORS                                  10

// Functions declarations
void HandleErrorDBC (SQLHDBC  hdbc1);
long NURand();
void LoadItem();
void LoadWarehouse();
void Stock();
void District();
void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
            double                                      ol;
   long          ol_i_id;
            long
            ol_supply_w_id;
   short         ol_quantity;
   double         ol_amount;
   char          ol_dist_info[DIST_INFO_LEN+1];
            char
            ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
   long          o_id;
   short         o_d_id;
            long                                        o_w_id;
   long          o_c_id;
   short         o_carrier_id;
```

```
   short          o_ol_cnt;
   short          o_all_local;
          ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
   long                                          c_id;
   short                                         c_d_id;
          long                                                c_w_id;
          char
          c_first[FIRST_NAME_LEN+1];
   char
          c_middle[MIDDLE_NAME_LEN+1];
   char
          c_last[LAST_NAME_LEN+1];
   char
          c_street_1[ADDRESS_LEN+1];
   char
          c_street_2[ADDRESS_LEN+1];
   char
          c_city[ADDRESS_LEN+1];
   char
          c_state[STATE_LEN+1];
   char                                          c_zip[ZIP_LEN+1];
   char
          c_phone[PHONE_LEN+1];
   char
          c_credit[CREDIT_LEN+1];
   double                                        c_credit_lim;
   double                                        c_discount;
          char
          c_balance[6];
   double                                        c_ytd_payment;
          short
          c_payment_cnt;
          short
          c_delivery_cnt;
          char
          c_data[C_DATA_LEN+1];
          double
          h_amount;
   char
          h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
          char
          c_last[LAST_NAME_LEN+1];
          char
          c_first[FIRST_NAME_LEN+1];
} CUSTOMER_SORT_STRUCT;                         c_id;

typedef struct
{
   long           time_start;
} LOADER_TIME_STRUCT;


// Global variables
char       szLastError[300];

HENV       henv;

HDBC       v_hdbc;
                  // for SQL Server version verification
HDBC       i_hdbc1;
           // for ITEM table
HDBC       w_hdbc1;
           // for WAREHOUSE, DISTRICT, STOCK
HDBC       c_hdbc1;
           // for CUSTOMER
HDBC       c_hdbc2;
           // for HISTORY
HDBC       o_hdbc1;
           // for ORDERS
HDBC       o_hdbc2;
           // for NEW-ORDER

HDBC       o_hdbc3;
           // for ORDER-LINE

HSTMT      v_hstmt;
           // for SQL Server version verification
HSTMT      i_hstmt1;
HSTMT      w_hstmt1;
HSTMT      c_hstmt1, c_hstmt2;
HSTMT      o_hstmt1, o_hstmt2, o_hstmt3;

long                  total_db_errors = 0;

ORDERS_STRUCT      orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT
          customer_buf[CUSTOMERS_PER_DISTRICT];
long                              orders_rows_loaded;
double                            new_order_rows_loaded;
double                            order_line_rows_loaded;
long                              history_rows_loaded;
long                              customer_rows_loaded;
double                            stock_rows_loaded;
long                              district_rows_loaded;
long                              item_rows_loaded;
long                              warehouse_rows_loaded;
long                              main_time_start;
long                              main_time_end;
long                              customers_per_district;
long                              orders_per_district;
long                              first_new_order;
long                              last_new_order;

TPCCLDR_ARGS    *aptr, args;

//=======================================
==============================
//
// Function name: main
//
//=======================================
==============================
int main(int  argc, char **argv)
{

          DWORD         dwThreadID[MAX_MAIN_THREADS];
          HANDLE        hThread[MAX_MAIN_THREADS];
          FILE          *fLoader;
          char          buffer[255];
          int                                         i;

          for (i=0; i<MAX_MAIN_THREADS; i++)
                    hThread[i] = NULL;


printf("\n*********************************************
*****");
          printf("\n*                                   *");
          printf("\n*  Microsoft SQL Server                 *");
          printf("\n*                                   *");
          printf("\n*  TPC-C BENCHMARK KIT:  Database loader
*");
          printf("\n*  Version %s                        *",
TPCKIT_VER);
          printf("\n*                                   *");
          printf("\n*********************************
**************\n\n");

   // process command line arguments
   aptr = &args;
   GetArgsLoader(argc, argv, aptr);

          printf("Build interface is ODBC.\n");

          if (aptr->build_index == 0)
                    printf("Data load only - no index
creation.\n");
          else
                    {
                    printf("Data load and index creation.\n");
```

```
                        if (aptr->index_order == 0)
                                printf("Clustered indexes will be
created after bulk load.\n");
                        else
                                printf("Clustered indexes will be
created before bulk load.\n");
                        }


        if (aptr->index_order == 0)
                printf("Clustered indexes will be created
after bulk load.\n");
        else
                printf("Clustered indexes will be created
before bulk load.\n");

        // set database scale values
        if (aptr->scale_down == 1)
        {
                printf("*** Scaled Down Database ***\n");
                max_items = MAXITEMS_SCALE_DOWN;
                customers_per_district =
CUSTOMERS_SCALE_DOWN;
                orders_per_district =
ORDERS_SCALE_DOWN;
                first_new_order = 0;
                last_new_order  = 30;
        }
        else
        {
                max_items = MAXITEMS;
                customers_per_district =
CUSTOMERS_PER_DISTRICT;
                orders_per_district =
ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }

        // open connections to SQL Server
        OpenConnections();

        // open file for loader results
        fLoader = fopen(aptr->loader_res_file, "w");


   if (fLoader == NULL)
        {
                printf("Error, loader result file open
failed.");
                exit(-1);
        }

   // start loading data
   sprintf(buffer,"TPC-C load started for %ld warehouses. starting
warehouse =%ld\n",(aptr->num_warehouses-aptr-
>starting_warehouse+1),
                        aptr->starting_warehouse);
        if      (aptr->scale_down == 1)
        {
                sprintf(buffer,"SCALED DOWN
DATABASE.\n");
        }

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads
        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads
for: item\n");


                hThread[0] = CreateThread(NULL,

                        0,

                        (LPTHREAD_START_ROUTINE)
LoadItem,

                        NULL,

                        0,

                        &dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating
creating thread = 0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for:
warehouse\n");

                hThread[1] = CreateThread(NULL,

                        0,

                        (LPTHREAD_START_ROUTINE)
LoadWarehouse,

                        NULL,

                        0,

                        &dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating
creating thread = 1.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_customer)
        {
                fprintf(fLoader, "Starting loader threads for:
customer\n");

                hThread[2] = CreateThread(NULL,

                        0,

                        (LPTHREAD_START_ROUTINE)
LoadCustomer,

                        NULL,

                        0,

                        &dwThreadID[2]);

                if (hThread[2] == NULL)
                {
                        printf("Error, failed in creating
creating main thread = 2.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_orders)
        {
                fprintf(fLoader, "Starting loader threads for:
orders\n");

                hThread[3] = CreateThread(NULL,

                        0,
```

```
                                (LPTHREAD_START_ROUTINE)
LoadOrders,

                                NULL,

                                0,

                                &dwThreadID[3]);

                        if (hThread[3] == NULL)
                        {
                                printf("Error, failed in creating
creating main thread = 3.\n");
                                exit(-1);
                        }
                }

                // Wait for threads to finish...
                for (i=0; i<MAX_MAIN_THREADS; i++)
                {
                        if (hThread[i] != NULL)
                        {
                                WaitForSingleObject(
hThread[i], INFINITE );
                                CloseHandle(hThread[i]);
                                hThread[i] = NULL;
                        }
                }

                main_time_end = (TimeNow() / MILLI);

   sprintf(buffer,"\nTPC-C load completed successfully in %ld
minutes.\n",
                                (main_time_end -
main_time_start)/60);

                printf("%s",buffer);
                fprintf(fLoader, "%s", buffer);

                fclose(fLoader);

                SQLFreeEnv(henv);

   exit(0);

                return 0;
}


//=======================================
==============================
```

```
// Function name: LoadItem
//
//=======================================
==============================
void LoadItem()
{
   int                          i;
                long            i_id;
                long            i_im_id;
   char                 i_name[I_NAME_LEN+1];
   double               i_price;
   char                 i_data[I_DATA_LEN+1];
                char            name[20];
                long            time_start;
                RETCODE         rc;
                DBINT           rcint;
                char            bcphint[128];
                char            err_log_path[256];

                // Seed with unique number
                seed(1);

                printf("Loading item table...\n");

                //if build index before load
                if ((aptr->build_index == 1) && (aptr->index_order
== 1))
                        BuildIndex("idxitmcl");

                InitString(i_name, I_NAME_LEN+1);
                InitString(i_data, I_DATA_LEN+1);

                sprintf(name, "%s..%s", aptr->database, "item");

                strcpy(err_log_path,aptr->log_path);
                strcat(err_log_path,"item.err");
                rc = bcp_init(i_hdbc1, name, NULL, err_log_path ,
DB_IN);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                if ((aptr->build_index == 1) && (aptr->index_order
== 1))
                {
                        sprintf(bcphint, "tablock,
ROWS_PER_BATCH = 100000");
                        rc = bcp_control(i_hdbc1, BCPHINTS,
(void*) bcphint);
                        if (rc != SUCCEED)
                                HandleErrorDBC(i_hdbc1);
                }

                i = 0;
```

```
                rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
                rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0,
I_NAME_LEN, NULL, 0, 0, ++i);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
                rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
                rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
                rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                //time_start = (TimeNow() / MILLI);
                time_start = TimeNow();

                item_rows_loaded = 0;

                for (i_id = 1; i_id <= max_items; i_id++)
                {
                        i_im_id = RandomNumber(1L, 10000L);

                        MakeAlphaStringPadded(14, 24,
I_NAME_LEN, i_name);

                        i_price = ((float) RandomNumber(100L,
10000L))/100.0;

                        MakeOriginalAlphaString(26, 50,
I_DATA_LEN, i_data, 10);

                        rc = bcp_sendrow(i_hdbc1);

                        if (rc != SUCCEED)
                                HandleErrorDBC(i_hdbc1);

                        item_rows_loaded++;
                        CheckForCommit(i_hdbc1, i_hstmt1,
item_rows_loaded, "item", &time_start);
                }

                rcint = bcp_done(i_hdbc1);
                if (rcint < 0)
                        HandleErrorDBC(i_hdbc1);
```

```
        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order
== 0))
                BuildIndex("idxitmcl");

}

//=======================================
=================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as
Warehouses are created
//
//=======================================
=================================
void LoadWarehouse()
{
        int                     i;
        long        w_id;
  char      w_name[W_NAME_LEN+1];
  char      w_street_1[ADDRESS_LEN+1];
  char      w_street_2[ADDRESS_LEN+1];
  char      w_city[ADDRESS_LEN+1];
  char      w_state[STATE_LEN+1];
  char      w_zip[ZIP_LEN+1];
  double    w_tax;
  double    w_ytd;
        char        name[20];
        long        time_start;
        RETCODE     rc;
        DBINT       rcint;
        char        bcphint[128];
        char        err_log_path[256];

        // Seed with unique number
        seed(2);

        printf("Loading warehouse table...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
                BuildIndex("idxwarcl");

        InitAddress(w_street_1, w_street_2, w_city, w_state,
w_zip);

        sprintf(name, "%s..%s", aptr->database,
"warehouse");

        //strcpy(err_log_path,aptr->log_path);
        //strcat(err_log_path,"whouse.err");
        sprintf(err_log_path,"%swhouse%d.err",aptr-
>log_path,aptr->starting_warehouse);
        printf("warehouse error file =%s\n",err_log_path);
        rc = bcp_init(w_hdbc1, name, NULL, err_log_path,
DB_IN);

        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        /*printf("Setting Tablock for Warehouse handle\n");
        rc = bcp_control(w_hdbc1, BCPHINTS, "tablock");
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
*/
        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
        {
                sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %d", aptr->num_warehouses);
                rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0,
W_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN,
NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        //time_start = (TimeNow() / MILLI);
        time_start = TimeNow();

        warehouse_rows_loaded = 0;

        for (w_id = (long)aptr->starting_warehouse; w_id <=
(long) aptr->num_warehouses; w_id++)
        {
                MakeAlphaStringPadded(6,10,
W_NAME_LEN, w_name);

                MakeAddress(w_street_1, w_street_2,
w_city, w_state, w_zip);

                w_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

                w_ytd = 300000.00;

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);

                warehouse_rows_loaded++;
                CheckForCommit(w_hdbc1, i_hstmt1,
warehouse_rows_loaded, "warehouse", &time_start);
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading warehouse table.\n");

        // if build index after load...
```

```
        if ((aptr->build_index == 1) && (aptr->index_order
== 0))
                BuildIndex("idxwarcl");

        //stock_rows_loaded = 0;
        stock_rows_loaded = 4294960000;
        district_rows_loaded = 0;

        District();
        Stock();
}

//=======================================
==============================
//
// Function   : District
//
//=======================================
==============================
void District()
{
   int              i;
        short       d_id;
        long        d_w_id;
   char     d_name[D_NAME_LEN+1];
   char     d_street_1[ADDRESS_LEN+1];
   char     d_street_2[ADDRESS_LEN+1];
   char     d_city[ADDRESS_LEN+1];
   char     d_state[STATE_LEN+1];
   char     d_zip[ZIP_LEN+1];
   double   d_tax;
   double   d_ytd;
        char        name[20];
   long     d_next_o_id;
        long        time_start;
        long        w_id;
        RETCODE     rc;
        DBINT       rcint;
        char        bcphint[128];
        char        err_log_path[256];

        // Seed with unique number
        seed(4);

        printf("Loading district table...\n");

        // build index before load
        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
                BuildIndex("idxdiscl");

        InitString(d_name, D_NAME_LEN+1);
        InitAddress(d_street_1, d_street_2, d_city, d_state,
d_zip);      sprintf(name, "%s..%s", aptr->database, "district");


//      strcpy(err_log_path,aptr->log_path);
//      strcat(err_log_path,"district.err");
        sprintf(err_log_path,"%sdistrict%d.err",aptr-
>log_path,aptr->starting_warehouse);

        rc = bcp_init(w_hdbc1, name, NULL, err_log_path,
DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
        {
                sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10));
                rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0,
D_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN,
NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        d_ytd  = 30000.0;

        d_next_o_id = orders_per_district+1;

        //time_start = (TimeNow() / MILLI);
        time_start = TimeNow();

        for (w_id = (long) aptr->starting_warehouse; w_id <=
(long) aptr->num_warehouses; w_id++)
        {
                d_w_id = w_id;

                for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
                {

        MakeAlphaStringPadded(6,10,D_NAME_LEN, d_name);

                        MakeAddress(d_street_1,
d_street_2, d_city, d_state, d_zip);

                        d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)

        HandleErrorDBC(w_hdbc1);

                        district_rows_loaded++;
                        CheckForCommit(w_hdbc1,
w_hstmt1, district_rows_loaded, "district", &time_start);
                }
   }

        rcint = bcp_done(w_hdbc1);
```

```
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);


        printf("Finished loading district table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order
== 0))
                BuildIndex("idxdiscl");

  return;
}

//=======================================
==============================
//
// Function   : Stock
//
//=======================================
==============================
void Stock()
{
        int                     i;
        long        s_i_id;
        long        s_w_id;
        float   PercentDone;
  short     s_quantity;
  char      s_dist_01[S_DIST_LEN+1];
  char      s_dist_02[S_DIST_LEN+1];
  char      s_dist_03[S_DIST_LEN+1];
  char      s_dist_04[S_DIST_LEN+1];
  char      s_dist_05[S_DIST_LEN+1];
  char      s_dist_06[S_DIST_LEN+1];
  char      s_dist_07[S_DIST_LEN+1];
  char      s_dist_08[S_DIST_LEN+1];
  char      s_dist_09[S_DIST_LEN+1];
  char      s_dist_10[S_DIST_LEN+1];
  long      s_ytd;
  short     s_order_cnt;
  short     s_remote_cnt;
  char      s_data[S_DATA_LEN+1];
  short     len;
        char        name[20];
        long        time_start;
        RETCODE   rc;
        DBINT       rcint;
        char        bcphint[128];
        char        err_log_path[256];

        // Seed with unique number
        seed(3);

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
                BuildIndex("idxstkcl");

        sprintf(name, "%s..%s", aptr->database, "stock");

        //rc = bcp_init(w_hdbc1, name, NULL,
"logs\\stock.err", DB_IN);
//      strcpy(err_log_path,aptr->log_path);
//      strcat(err_log_path,"stock.err");
        sprintf(err_log_path,"%sstock%d.err",aptr-
>log_path,aptr->starting_warehouse);

        rc = bcp_init(w_hdbc1, name, NULL, err_log_path,
DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
        {
                sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 100000));
                rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt = 0;

        //time_start = (TimeNow() / MILLI);
        time_start = TimeNow();

        printf("...Loading stock table\n");
```

```
		for (s_i_id=1; s_i_id <= max_items; s_i_id++)
		{

			for (s_w_id = (long)aptr->starting_warehouse; s_w_id <= (long) aptr->num_warehouses; s_w_id++)
			{

				s_quantity = (short)RandomNumber(10L,100L);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
				len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

				len = MakeOriginalAlphaString(26,50, S_DATA_LEN, s_data,10);

				rc = bcp_sendrow(w_hdbc1);
				if (rc != SUCCEED)

		HandleErrorDBC(w_hdbc1);

				stock_rows_loaded++;
				CheckForCommit_Big(w_hdbc1, w_hstmt1, stock_rows_loaded, "stock", &time_start);

			} //display progress every 10%
//			if (!(fmod(s_i_id,10)) {

			if ((s_i_id % 10) == 0) {
				PercentDone = (float) s_i_id *100.0 / (float) max_items;
				printf("Processed Item %ld, Percent Done = %.2f\n",s_i_id,PercentDone);
				}
```

```
		}
		rcint = bcp_done(w_hdbc1);
		if (rcint < 0)
			HandleErrorDBC(w_hdbc1);


		printf("Finished loading stock table.\n");

		SQLFreeStmt(w_hstmt1, SQL_DROP);
		SQLDisconnect(w_hdbc1);
		SQLFreeConnect(w_hdbc1);

		// if build index after load...
		if ((aptr->build_index == 1) && (aptr->index_order == 0))
			BuildIndex("idxstkcl");

		return;
}

//=======================================================================
//
// Function   : LoadCustomer
//
//=======================================================================
void LoadCustomer()
{
		LOADER_TIME_STRUCT	customer_time_start;
		LOADER_TIME_STRUCT	history_time_start;
		long
		w_id;
  short			d_id;
		DWORD
		dwThreadID[MAX_CUSTOMER_THREADS];
		HANDLE
		hThread[MAX_CUSTOMER_THREADS];
		char			name[20];
		RETCODE
		rc;
		DBINT
		rcint;
		char
		bcphint[128];
		char
		cmd[256];
		int
			num_procs;
		char
		err_log_path_cust[256];
		char
		err_log_path_hist[256];
```

```
		// Seed with unique number
		seed(5);

		printf("Loading customer and history tables...\n");

		// if build index before load...
		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			BuildIndex("idxcuscl");
			 // check the number of processors on this system
			 // if 8 or more processors, then build index on History.
			 // if less than 8 processors, do not build the index
			num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
			if ( num_procs >= 8 )
				BuildIndex("idxhiscl");
		}

		// Initialize bulk copy
		sprintf(name, "%s..%s", aptr->database, "customer");


		//rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", DB_IN);
//		strcpy(err_log_path_cust,aptr->log_path);
//		strcat(err_log_path_cust,"customer.err");
		sprintf(err_log_path_cust,"%scustomer%d.err",aptr->log_path,aptr->starting_warehouse);
		rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust, DB_IN);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);

		if ((aptr->build_index == 1) && (aptr->index_order == 1))
		{
			sprintf(bcphint, "tablock, ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
			rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
			if (rc != SUCCEED)
				HandleErrorDBC(c_hdbc1);
		}

		sprintf(name, "%s..%s", aptr->database, "history");

//		rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
//		strcpy(err_log_path_hist,aptr->log_path);
```

```
//          strcat(err_log_path_hist,"history.err");
            sprintf(err_log_path_cust,"%shistory%d.err",aptr-
>log_path,aptr->starting_warehouse);
            rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist,
DB_IN);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

            sprintf(bcphint, "tablock");
            rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
            if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

            customer_rows_loaded    = 0;
            history_rows_loaded     = 0;

            CustomerBufInit();

            customer_time_start.time_start = (TimeNow() /
MILLI);
            history_time_start.time_start = (TimeNow() / MILLI);

            for (w_id = (long)aptr->starting_warehouse; w_id <=
(long) aptr->num_warehouses; w_id++)
            {
                        for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
                        {

                                    CustomerBufLoad(d_id, w_id);

                                    // Start parallel loading threads
here...
                                    // Start customer table thread
                                    printf("...Loading customer
table for: d_id = %ld, w_id = %ld\n", d_id, w_id);

                                    hThread[0] =
CreateThread(NULL,

                                                0,


(LPTHREAD_START_ROUTINE) LoadCustomerTable,


&customer_time_start,

                                                0,

                                                &dwThreadID[0]);

                                    if (hThread[0] == NULL)
                                                printf("Error, failed
in creating creating thread = 0.\n");
                                                exit(-1);
                                    }

                                    // Start History table thread
                                    printf("...Loading history table
for: d_id = %ld, w_id = %ld\n", d_id, w_id);

                                    hThread[1] =
CreateThread(NULL,

                                                0,


(LPTHREAD_START_ROUTINE) LoadHistoryTable,


&history_time_start,

                                                0,

                                                &dwThreadID[1]);

                                    if (hThread[1] == NULL)
                                    {
                                                printf("Error, failed
in creating creating thread = 1.\n");
                                                exit(-1);
                                    }

                                    WaitForSingleObject(
hThread[0], INFINITE );
                                    WaitForSingleObject(
hThread[1], INFINITE );

                                    if (CloseHandle(hThread[0]) ==
FALSE)
                                    {
                                                printf("Error, failed
in closing customer thread handle with errno: %d\n",
GetLastError());
                                    }

                                    if (CloseHandle(hThread[1]) ==
FALSE)
                                    {
                                                printf("Error, failed
in closing history thread handle with errno: %d\n", GetLastError());
                                    }
                        }
            }

            // flush the bulk connection
            rcint = bcp_done(c_hdbc1);
            if (rcint < 0)
                        HandleErrorDBC(c_hdbc1);


            rcint = bcp_done(c_hdbc2);
            if (rcint < 0)
                        HandleErrorDBC(c_hdbc2);


            printf("Finished loading customer table.\n");

            // if build index after load...
            if ((aptr->build_index == 1) && (aptr->index_order
== 0))
            {
                        BuildIndex("idxcuscl");
                        // check the number of processors on this
system
                        // if 8 or more processors, then build index
on History.
                        // if less than 8 processors, do not build the
index
                        num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
                        if (num_procs >= 8)
                                    BuildIndex("idxhiscl");
            }

            // build non-clustered index
            if (aptr->build_index == 1)
                        BuildIndex("idxcusnc");

            // Output the NURAND used for the loader into C_FIRST
for C_ID = 1,
            // C_W_ID = 1, and C_D_ID = 1
/*          sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -
Q\"update customer set c_first = 'C_LOAD = %d' where c_id = 1
and c_w_id = 1 and c_d_id = 1\" > %snurand_load.log",
                                                aptr->server,
                                                aptr->user,
                                                aptr->password,
                                                aptr->database,

            LOADER_NURAND_C,
                                                aptr->log_path);

            system(cmd);
*/

            SQLFreeStmt(c_hstmt1, SQL_DROP);
            SQLDisconnect(c_hdbc1);
```

```
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);

  return;
}

//=======================================
==============================
//
// Function   : CustomerBufInit
//
//=======================================
==============================
void CustomerBufInit()
{
        long    i;

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_id = 0;
                customer_buf[i].c_d_id = 0;
                customer_buf[i].c_w_id = 0;

                strcpy(customer_buf[i].c_first,"");
                strcpy(customer_buf[i].c_middle,"");
                strcpy(customer_buf[i].c_last,"");
                strcpy(customer_buf[i].c_street_1,"");
                strcpy(customer_buf[i].c_street_2,"");
                strcpy(customer_buf[i].c_city,"");
                strcpy(customer_buf[i].c_state,"");
                strcpy(customer_buf[i].c_zip,"");
                strcpy(customer_buf[i].c_phone,"");
                strcpy(customer_buf[i].c_credit,"");

                customer_buf[i].c_credit_lim = 0;
                customer_buf[i].c_discount = (float) 0;

                strcpy(customer_buf[i].c_balance,"");

                customer_buf[i].c_ytd_payment = 0;
                customer_buf[i].c_payment_cnt = 0;
                customer_buf[i].c_delivery_cnt = 0;

                strcpy(customer_buf[i].c_data,"");


                customer_buf[i].h_amount = 0;

                strcpy(customer_buf[i].h_data,"");
        }
}

//=======================================
==============================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=======================================
==============================
void CustomerBufLoad(int d_id, long w_id)
{
        long                                    i;
  CUSTOMER_SORT_STRUCT  c[CUSTOMERS_PER_DISTRICT];

        for (i=0;i<customers_per_district;i++)
        {
                if (i < 1000)
                        LastName(i, c[i].c_last);
                else

        LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);


        MakeAlphaStringPadded(8,16,FIRST_NAME_LEN,
c[i].c_first);

                c[i].c_id = i+1;
        }

        printf("...Loading customer buffer for: d_id = %ld, w_id
= %ld\n",
                        d_id, w_id);

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_d_id = d_id;
                customer_buf[i].c_w_id = w_id;
                customer_buf[i].h_amount = 10.0;
                customer_buf[i].c_ytd_payment  = 10.0;

                customer_buf[i].c_payment_cnt  = 1;
                customer_buf[i].c_delivery_cnt = 0;
                customer_buf[i].c_id = c[i].c_id;
                strcpy(customer_buf[i].c_first, c[i].c_first);
                strcpy(customer_buf[i].c_last, c[i].c_last);
                customer_buf[i].c_middle[0] = 'O';
                customer_buf[i].c_middle[1] = 'E';
                MakeAddress(customer_buf[i].c_street_1,

customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,

customer_buf[i].c_zip);
                MakeNumberString(16, 16, PHONE_LEN,
customer_buf[i].c_phone);

                if (RandomNumber(1L, 100L) > 10)
                        customer_buf[i].c_credit[0] =
'G';
                else
                        customer_buf[i].c_credit[0] =
'B';
                customer_buf[i].c_credit[1] = 'C';
                customer_buf[i].c_credit_lim = 50000.0;

                customer_buf[i].c_discount = ((float)
RandomNumber(0L, 5000L)) / 10000.0;

                strcpy(customer_buf[i].c_balance,"-10.0");
                MakeAlphaStringPadded(300, 500,
C_DATA_LEN, customer_buf[i].c_data);

                // Generate HISTORY data
                MakeAlphaStringPadded(12, 24,
H_DATA_LEN, customer_buf[i].h_data);
        }
}

//=======================================
==============================
//
// Function   : LoadCustomerTable
//
//=======================================
==============================
void LoadCustomerTable(LOADER_TIME_STRUCT
*customer_time_start)
{
        long                    i;
  long      c_id;
  short     c_d_id;
        long                    c_w_id;
  char      c_first[FIRST_NAME_LEN+1];
  char      c_middle[MIDDLE_NAME_LEN+1];
  char      c_last[LAST_NAME_LEN+1];
  char      c_street_1[ADDRESS_LEN+1];
  char      c_street_2[ADDRESS_LEN+1];
  char      c_city[ADDRESS_LEN+1];
  char      c_state[STATE_LEN+1];
  char      c_zip[ZIP_LEN+1];
  char      c_phone[PHONE_LEN+1];
```

```
   char      c_credit[CREDIT_LEN+1];
   double    c_credit_lim;
   double    c_discount;
             char                    c_balance[6];
   double    c_ytd_payment;
   short     c_payment_cnt;
   short     c_delivery_cnt;
   char      c_data[C_DATA_LEN+1];
             char                    c_since[C_SINCE_LEN+1];

             RETCODE                 rc;

   i = 0;
             rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
             rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN,
NULL, 0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
             rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0,
FIRST_NAME_LEN, NULL, 0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL,
0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
             if (rc != SUCCEED)
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN,
NULL, 0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN,
NULL,0,0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL,
0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0,
0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0,
++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL,
0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
             rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0,
C_SINCE_LEN, NULL, 0, SQLCHARACTER, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
             rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0,
MIDDLE_NAME_LEN,NULL, 0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);
   rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, C_DATA_LEN, NULL,
0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc1);

   for (i = 0; i < customers_per_district; i++)
   {
                       c_id = customer_buf[i].c_id;
                       c_d_id = customer_buf[i].c_d_id;
                       c_w_id = customer_buf[i].c_w_id;

                       strcpy(c_first, customer_buf[i].c_first);
                       strcpy(c_middle,
customer_buf[i].c_middle);
                       strcpy(c_last, customer_buf[i].c_last);
                       strcpy(c_street_1,
customer_buf[i].c_street_1);
                       strcpy(c_street_2,
customer_buf[i].c_street_2);
                       strcpy(c_city, customer_buf[i].c_city);
                       strcpy(c_state, customer_buf[i].c_state);
                       strcpy(c_zip, customer_buf[i].c_zip);
                       strcpy(c_phone, customer_buf[i].c_phone);
                       strcpy(c_credit, customer_buf[i].c_credit);

                       FormatDate(&c_since);


                       c_credit_lim =
customer_buf[i].c_credit_lim;
                       c_discount = customer_buf[i].c_discount;
                       strcpy(c_balance,
customer_buf[i].c_balance);
                       c_ytd_payment =
customer_buf[i].c_ytd_payment;
                       c_payment_cnt =
customer_buf[i].c_payment_cnt;
                       c_delivery_cnt =
customer_buf[i].c_delivery_cnt;
                       strcpy(c_data, customer_buf[i].c_data);


                       // Send data to server
                       rc = bcp_sendrow(c_hdbc1);
                       if (rc != SUCCEED)
                                 HandleErrorDBC(c_hdbc1);


                       customer_rows_loaded++;
                       CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start-
>time_start);
   }
}

//=======================================
==============================
//
// Function   : LoadHistoryTable
//
//=======================================
==============================
void LoadHistoryTable(LOADER_TIME_STRUCT
*history_time_start)
{
             long                    i;
```

```
   long      c_id;
   short     c_d_id;
             long                    c_w_id;
             double      h_amount;
   char      h_data[H_DATA_LEN+1];
             char                    h_date[H_DATE_LEN+1];

             RETCODE                 rc;

             i = 0;
   rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
             rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
   rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
             rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
             rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0,
H_DATE_LEN, NULL, 0, SQLCHARACTER, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
   rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);
   rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL,
0, 0, ++i);
             if (rc != SUCCEED)
                       HandleErrorDBC(c_hdbc2);

   for (i = 0; i < customers_per_district; i++)
   {
                       c_id = customer_buf[i].c_id;
                       c_d_id = customer_buf[i].c_d_id;
                       c_w_id = customer_buf[i].c_w_id;
                       h_amount = customer_buf[i].h_amount;
                       strcpy(h_data, customer_buf[i].h_data);
                       FormatDate(&h_date);

                       // send to server
                       rc = bcp_sendrow(c_hdbc2);
                       if (rc != SUCCEED)

                                 HandleErrorDBC(c_hdbc2);


                       history_rows_loaded++;
                       CheckForCommit(c_hdbc2, c_hstmt2,
history_rows_loaded, "history", &history_time_start->time_start);
   }
}

//=======================================
==================-===============
//
// Function   : LoadOrders
//
//=======================================
==================================
void LoadOrders()
{
             LOADER_TIME_STRUCT     orders_time_start;
             LOADER_TIME_STRUCT     new_order_time_start;
             LOADER_TIME_STRUCT     order_line_time_start;
             long
             w_id;
   short                   d_id;
             DWORD
             dwThreadID[MAX_ORDER_THREADS];
             HANDLE
             hThread[MAX_ORDER_THREADS];
             char                    name[20];
             RETCODE
             rc;
             char
             bcphint[128];
             char
             err_log_path_ord[256];
             char
             err_log_path_nord[256];
             char
             err_log_path_ordl[256];

             // seed with unique number
             seed(6);

             printf("Loading orders...\n");

             // if build index before load...
             if ((aptr->build_index == 1) && (aptr->index_order
== 1))      {
                                 BuildIndex("idxordcl");
                                 BuildIndex("idxnodcl");
                                 BuildIndex("idxodlcl");
             }

             // initialize bulk copy
             sprintf(name, "%s..%s", aptr->database, "orders");

//           strcpy(err_log_path_ord,aptr->log_path);
//           strcat(err_log_path_ord,"orders.err");
             sprintf(err_log_path_ord,"%sorders%d.err",aptr-
>log_path,aptr->starting_warehouse);
             rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord,
DB_IN);
             if (rc != SUCCEED)
                       HandleErrorDBC(o_hdbc1);

             if ((aptr->build_index == 1) && (aptr->index_order
== 1))
             {
                       sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
                       rc = bcp_control(o_hdbc1, BCPHINTS,
(void*) bcphint);
                       if (rc != SUCCEED)
                                 HandleErrorDBC(o_hdbc1);
             }

             sprintf(name, "%s..%s", aptr->database,
"new_order");

//           rc = bcp_init(o_hdbc2, name, NULL,
"logs\\neword.err", DB_IN);
//           strcpy(err_log_path_nord,aptr->log_path);
//           strcat(err_log_path_nord,"neword.err");
             sprintf(err_log_path_ord,"%sneword%d.err",aptr-
>log_path,aptr->starting_warehouse);
             rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord,
DB_IN);
             if (rc != SUCCEED)
                       HandleErrorDBC(o_hdbc2);

             if ((aptr->build_index == 1) && (aptr->index_order
== 1))
             {
                       sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
                       rc = bcp_control(o_hdbc2, BCPHINTS,
(void*) bcphint);
                       if (rc != SUCCEED)
                                 HandleErrorDBC(o_hdbc2);
```

```
        }

        sprintf(name, "%s..%s", aptr->database, "order_line");

        //      rc = bcp_init(o_hdbc3, name, NULL,
"logs\\ordline.err", DB_IN);
//      strcpy(err_log_path_ordl,aptr->log_path);
//      strcat(err_log_path_ordl,"ordline.err");
        sprintf(err_log_path_ordl,"%sordline%d.err",aptr-
>log_path,aptr->starting_warehouse);
        rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl,
DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

   if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
                rc = bcp_control(o_hdbc3, BCPHINTS,
(void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
        }

        orders_rows_loaded     = 0;
        new_order_rows_loaded  = 0;
        //order_line_rows_loaded  = 0;
        order_line_rows_loaded          = 4294960000;

        OrdersBufInit();

        orders_time_start.time_start = (TimeNow() / MILLI);
        new_order_time_start.time_start = (TimeNow() /
MILLI);
        order_line_time_start.time_start = (TimeNow() /
MILLI);

        for (w_id = (long)aptr->starting_warehouse; w_id <=
(long) aptr->num_warehouses; w_id++)
        {
                for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
                {

                        OrdersBufLoad(d_id, w_id);

                        // start parallel loading threads
here...
                        // start Orders table thread
                        printf("...Loading Order Table
for: d_id = %ld, w_id = %ld\n", d_id, w_id);

                                hThread[0] =
CreateThread(NULL,
                                        0,

(LPTHREAD_START_ROUTINE) LoadOrdersTable,

&orders_time_start,
                                        0,
                                        &dwThreadID[0]);
                                if (hThread[0] == NULL)
                                {
                                        printf("Error, failed
in creating creating thread = 0.\n");
                                        exit(-1);
                                }

                                // start NewOrder table thread
                                printf("...Loading New-Order
Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                                hThread[1] =
CreateThread(NULL,
                                        0,

(LPTHREAD_START_ROUTINE) LoadNewOrderTable,

&new_order_time_start,
                                        0,
                                        &dwThreadID[1]);
                                if (hThread[1] == NULL)
                                {
                                        printf("Error, failed
in creating creating thread = 1.\n");
                                        exit(-1);
                                }

                                // start Order-Line table thread
                                printf("...Loading Order-Line
Table for: d_id = %ld, w_id = %ld\n", d_id, w_id);

                                hThread[2] =
CreateThread(NULL,
                                        0,

(LPTHREAD_START_ROUTINE) LoadOrderLineTable,

&order_line_time_start,
                                        0,
                                        &dwThreadID[2]);
                                if (hThread[2] == NULL)
                                {
                                        printf("Error, failed
in creating creating thread = 2. LastError = %d\n",GetLastError());
                                        exit(-1);
                                }

                                WaitForSingleObject(
hThread[0], INFINITE );
                                WaitForSingleObject(
hThread[1], INFINITE );
                                WaitForSingleObject(
hThread[2], INFINITE );

                                if (CloseHandle(hThread[0]) ==
FALSE)
                                {
                                        printf("Error, failed
in closing Orders thread handle with errno: %d\n", GetLastError());
                                }

                                if (CloseHandle(hThread[1]) ==
FALSE)
                                {
                                        printf("Error, failed
in closing NewOrder thread handle with errno: %d\n",
GetLastError());
                                }

                                if (CloseHandle(hThread[2]) ==
FALSE)
                                {
                                        printf("Error, failed
in closing OrderLine thread handle with errno: %d\n",
GetLastError());
                                }
                        }
                }
```

```
		printf("Finished loading orders.\n");

	return;
}

//=========================================
================================
//
// Function	: OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//=========================================
================================
void OrdersBufInit()
{
		int	i;
		int			j;

		for (i=0;i<orders_per_district;i++)
		{
			orders_buf[i].o_id = 0;
			orders_buf[i].o_d_id = 0;
			orders_buf[i].o_w_id = 0;
			orders_buf[i].o_c_id = 0;
			orders_buf[i].o_carrier_id = 0;
			orders_buf[i].o_ol_cnt = 0;
			orders_buf[i].o_all_local = 0;

			for (j=0;j<=14;j++)
			{
				orders_buf[i].o_ol[j].ol = 0;
				orders_buf[i].o_ol[j].ol_i_id =
0;

		orders_buf[i].o_ol[j].ol_supply_w_id = 0;

		orders_buf[i].o_ol[j].ol_quantity = 0;
				orders_buf[i].o_ol[j].ol_amount
= 0;

		strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
			}
		}
}

//=========================================
================================
//
// Function	: OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//=========================================
================================
void OrdersBufLoad(short d_id, long w_id)
{

		int	cust[ORDERS_PER_DISTRICT+1];
		long	o_id;
		long		ol;

		printf("...Loading Order Buffer for: d_id = %ld, w_id =
%ld\n",
				d_id, w_id);

		GetPermutation(cust, orders_per_district);

		for (o_id=0;o_id<orders_per_district;o_id++)
		{
			// Generate ORDER and NEW-ORDER data
			orders_buf[o_id].o_d_id = d_id;
			orders_buf[o_id].o_w_id = w_id;
			orders_buf[o_id].o_id = o_id+1;
			orders_buf[o_id].o_c_id = cust[o_id+1];
			orders_buf[o_id].o_ol_cnt =
(short)RandomNumber(5L, 15L);

			if (o_id < first_new_order)
			{
				orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
				orders_buf[o_id].o_all_local  =
1;
			}
			else
			{
				orders_buf[o_id].o_carrier_id =
0;
				orders_buf[o_id].o_all_local  =
1;
			}

			for (ol=0; ol<orders_buf[o_id].o_ol_cnt;
ol++)
			{

				orders_buf[o_id].o_ol[ol].ol =
ol+1;

		orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L,
max_items);

		orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;

		orders_buf[o_id].o_ol[ol].ol_quantity = 5;
				MakeAlphaString(24, 24,
OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);
				// Generate ORDER-LINE data
				if (o_id < first_new_order)
				{

		orders_buf[o_id].o_ol[ol].ol_amount = 0;
					// Added to insure
ol_delivery_d set properly during load

		FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);


				}
				else
				{

		orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
					// Added to insure
ol_delivery_d set properly during load

					// odbc datetime
format

		strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-
12-31 00:00:00.000");
				}
			}
		}
}

//=========================================
================================
//
// Function	: LoadOrdersTable
//
//=========================================
================================
void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
		int	i;
	long	o_id;
		short	o_d_id;
		long		o_w_id;
	long	o_c_id;
	short	o_carrier_id;
	short	o_ol_cnt;
	short	o_all_local;
		char
		o_entry_d[O_ENTRY_D_LEN+1];
		RETCODE		rc;
		DBINT		rcint;
```

```
		// bind ORDER data
	i = 0;
		rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
		rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
	rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);
		rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0,
O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc1);

	for (i = 0; i < orders_per_district; i++)
	{
			o_id		= orders_buf[i].o_id;
			o_d_id		= orders_buf[i].o_d_id;
			o_w_id		= orders_buf[i].o_w_id;
			o_c_id		= orders_buf[i].o_c_id;
			o_carrier_id = orders_buf[i].o_carrier_id;
			o_ol_cnt	= orders_buf[i].o_ol_cnt;
			o_all_local	= orders_buf[i].o_all_local;


			FormatDate(&o_entry_d);

			// send data to server
			rc = bcp_sendrow(o_hdbc1);
			if (rc != SUCCEED)
				HandleErrorDBC(o_hdbc1);


			orders_rows_loaded++;
			CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start->time_start);
	}

		if ((o_w_id == aptr->num_warehouses) && (o_d_id
== 10))
		{
			rcint = bcp_done(o_hdbc1);

			if (rcint < 0)
				HandleErrorDBC(o_hdbc1);


			SQLFreeStmt(o_hstmt1, SQL_DROP);
			SQLDisconnect(o_hdbc1);
			SQLFreeConnect(o_hdbc1);

			// if build index after load...
			if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
				BuildIndex("idxordcl");

			// build non-clustered index
			if (aptr->build_index == 1)
				BuildIndex("idxordnc");
		}
}

//=======================================
==============================
//
// Function	: LoadNewOrderTable
//
//=======================================
==============================
void LoadNewOrderTable(LOADER_TIME_STRUCT
*new_order_time_start)
{
		long		i;
	long	o_id;
	short	o_d_id;
		long		o_w_id;
		RETCODE		rc;
		DBINT		rcint;

		// Bind NEW-ORDER data
		i = 0;
	rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);
	rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);
		rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(o_hdbc2);

		for (i = first_new_order; i < last_new_order; i++)
		{
			o_id	= orders_buf[i].o_id;
			o_d_id	= orders_buf[i].o_d_id;
			o_w_id	= orders_buf[i].o_w_id;

			rc = bcp_sendrow(o_hdbc2);
			if (rc != SUCCEED)

				HandleErrorDBC(o_hdbc2);


			new_order_rows_loaded++;

			CheckForCommit_Big(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start-
>time_start);
		}

		if ((o_w_id == aptr->num_warehouses) && (o_d_id
== 10))
		{
			rcint = bcp_done(o_hdbc2);

			if (rcint < 0)
				HandleErrorDBC(o_hdbc2);


			SQLFreeStmt(o_hstmt2, SQL_DROP);
			SQLDisconnect(o_hdbc2);
			SQLFreeConnect(o_hdbc2);

			// if build index after load...
			if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
				BuildIndex("idxnodcl");
		}
}

//=======================================
==============================
```

```
//
// Function   : LoadOrderLineTable
//
//=========================================
==============================
void LoadOrderLineTable(LOADER_TIME_STRUCT
*order_line_time_start)
{
            long                    i;
            long                    j;
   long     o_id;
            short       o_d_id;
            long                    o_w_id;
            double                  ol;
            long        ol_i_id;
            long                    ol_supply_w_id;
   short    ol_quantity;
   double    ol_amount;
   char     ol_dist_info[DIST_INFO_LEN+1];
            char
            ol_delivery_d[OL_DELIVERY_D_LEN+1];
            RETCODE                 rc;
            DBINT                   rcint;

            // bind ORDER-LINE data
            i = 0;
   rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
   rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
            rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
            rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
            rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
            rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
   rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);
   rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 0, ++i);
            if (rc != SUCCEED)
                        HandleErrorDBC(o_hdbc3);

            for (i = 0; i < orders_per_district; i++)
   {
                        o_id    = orders_buf[i].o_id;
                        o_d_id  = orders_buf[i].o_d_id;
                        o_w_id  = orders_buf[i].o_w_id;

                        for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                        {
                                    ol          =
orders_buf[i].o_ol[j].ol;
                                    ol_i_id       =
orders_buf[i].o_ol[j].ol_i_id;
                                    ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
                                    ol_quantity   =
orders_buf[i].o_ol[j].ol_quantity;
                                    ol_amount     =
orders_buf[i].o_ol[j].ol_amount;

            strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d
);


            strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                                    rc = bcp_sendrow(o_hdbc3);
                                    if (rc != SUCCEED)


            HandleErrorDBC(o_hdbc3);

                                    order_line_rows_loaded++;

                                    CheckForCommit_Big(o_hdbc3,
o_hstmt3, order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
                        }
            }
            if ((o_w_id == aptr->num_warehouses) && (o_d_id
== 10))    {
                        rcint = bcp_done(o_hdbc3);

                        if (rcint < 0)
                                    HandleErrorDBC(o_hdbc3);


                        SQLFreeStmt(o_hstmt3, SQL_DROP);
                        SQLDisconnect(o_hdbc3);
                        SQLFreeConnect(o_hdbc3);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
                                    BuildIndex("idxodlcl");
            }
}

//=========================================
==============================
//
// Function   : GetPermutation
//
//=========================================
==============================
void GetPermutation(int perm[], int n)
{
            int i, r, t;

            for (i=1;i<=n;i++)
                        perm[i] = i;

            for (i=1;i<=n;i++)
            {
                        r = RandomNumber(i,n);
                        t = perm[i];
                        perm[i] = perm[r];
                        perm[r] = t;
            }
}

//=========================================
==============================
//
// Function   : CheckForCommit
//
//=========================================
==============================
void CheckForCommit(HDBC hdbc,
                                                HSTMT
hstmt,
```

```
                                                long
rows_loaded,
                                                char
*table_name,
                                        long *time_start)
{

   long       time_end, time_diff;

   if ( !(rows_loaded % aptr->batch) )
   {
//                      time_end = (TimeNow() / MILLI);
                        time_end = TimeNow();
                        time_diff = time_end - *time_start;

                        printf("-> Loaded %ld rows into %s in %ld
ms - Total = %d (%.2f rpms)\n",
                                aptr->batch,
                                table_name,
                                        time_diff,
                                rows_loaded,
                                        (float) aptr->batch
/ (time_diff ? time_diff : 1L));

                        *time_start = time_end;
//                      bcp_batch(hdbc);
   }

   return;
}

//======================================
==============================
//
// Function   : CheckForCommit_Big
//
//======================================
==============================
void CheckForCommit_Big(HDBC hdbc,
                                                HSTMT
hstmt,
                                                double
rows_loaded,
                                                char
*table_name,
                                        long *time_start)
{

   long       time_end, time_diff;

   if ( !(fmod(rows_loaded,aptr->batch) ) )
   {
//                      time_end = (TimeNow() / MILLI);
                        time_end = TimeNow();
                        time_diff = time_end - *time_start;

                        printf("-> Loaded %ld rows into %s in %ld
ms - Total = %.0f (%.2f rpms)\n",
                                aptr->batch,
                                table_name,
                                        time_diff,
                                rows_loaded,
                                        (float) aptr->batch
/ (time_diff ? time_diff : 1L));

                        *time_start = time_end;
//                      bcp_batch(hdbc);
   }

   return;
}

//======================================
==============================
//
// Function   : OpenConnections
//
//======================================
==============================
void OpenConnections()
{

        RETCODE       rc;

        char                            szDriverString[300];
        char
        szDriverStringOut[1024];
        SQLSMALLINT                     cbDriverStringOut;


        SQLAllocHandle(SQL_HANDLE_ENV,
SQL_NULL_HA NDLE, &henv );

        SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

        SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
        SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

        SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );
        SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

        // Open connections to SQL Server
        // Connection 1
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );
        printf("opening db connections, open string =
%s\n",szDriverString);

        rc = SQLSetConnectOption (i_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        rc = SQLDriverConnect ( i_hdbc1,
                                                NULL,

        (SQLCHAR*)&szDriverString[0] ,

        SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),

        &cbDriverStringOut,

        SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(i_hdbc1);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }
```

```
        // Connection 2
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );

        rc = SQLSetConnectOption (w_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);

        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

        rc = SQLDriverConnect ( w_hdbc1,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );

        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(w_hdbc1);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 3
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );
        rc = SQLSetConnectOption (c_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        rc = SQLDriverConnect ( c_hdbc1,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(c_hdbc1);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 4
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );

        rc = SQLSetConnectOption (c_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        rc = SQLDriverConnect ( c_hdbc2,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,
                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(c_hdbc2);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 5
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );

        rc = SQLSetConnectOption (o_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

        rc = SQLDriverConnect ( o_hdbc1,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(o_hdbc1);
```

```
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 6
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,

         aptr->database );

        rc = SQLSetConnectOption (o_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(o_hdbc2);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 7
        sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

         aptr->server,

         aptr->user,

         aptr->password,
         aptr->database );

        rc = SQLSetConnectOption (o_hdbc3,
SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,

                NULL,

                (SQLCHAR*)&szDriverString[0] ,

                SQL_NTS,

                (SQLCHAR*)&szDriverStringOut[0],

                sizeof(szDriverStringOut),

                &cbDriverStringOut,

                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                 (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(o_hdbc3);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }
}

//=========================================
==============================
//
// Function name: BuildIndex
//
//=========================================
==============================
void BuildIndex(char    *index_script)
{
        char    cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "osql -S%s -U%s -P%s -e -i%s\\%s.sql >
%s%s.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr-
>index_script_path,
                                        aptr->log_path,
                                        index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}

//=========================================
==============================
//
// Function name: HandleErrorDBC
//
//=========================================
==============================
void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR         SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER      NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char            timebuf[128];
        char            datebuf[128];
        char            err_log_path[256];
        FILE            *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC ,
hdbc1, i, SqlState , &NativeError,
                                        Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
        {
                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n==>SQLState:
%s\n" , datebuf, timebuf, szLastError, SqlState);

//              strcpy(err_log_path,aptr->log_path);
//              strcat(err_log_path,"tpccldr.err");
        sprintf(err_log_path,"%stpccldr%d.err",aptr-
>log_path,aptr->starting_warehouse);
                fp1 = fopen(err_log_path,"a+");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open
errorlog file.\n");
                else
                {
```

```
                                fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
                                fclose(fp1);
                        }

                        i++;
                }
}

//=======================================
==============================
//
// Function   : HandleErrorSTMT
//
//=======================================
==============================
void HandleErrorSTMT (HSTMT  hstmt1)
{
        SQLCHAR             SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER          NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char                timebuf[128];
        char                datebuf[128];
        char                err_log_path[256];
        FILE                *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT ,
hstmt1, i, SqlState , &NativeError,
                                        Msg, sizeof(Msg) ,
&MsgLen )) != SQL_NO_DATA )
        {
                if (total_db_errors >= MAX_SQL_ERRORS)
                {
                        printf(">>>>> Maximum SQL
errors of %d exceeded.  Terminating
TPCCLDR.<<<<<\n",total_db_errors);
                        exit(9);
                }
                total_db_errors++;

                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\nSQLState: %s\n" ,
datebuf, timebuf, szLastError, SqlState);

//              strcpy(err_log_path,aptr->log_path);
//              sprintf(err_log_path,"%s/stock%d%d.err",aptr-
>log_path,aptr->starting_warehouse);
                strcat(err_log_path,"/tpcclderr.err");
                fp1 = fopen(err_log_path,"a+");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open
errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);
                        fclose(fp1);
                }

                i++;
        }
}

//=======================================
==============================
//
// Function   : FormatDate
//
//=======================================
==============================
void FormatDate ( char* szTimeCOutput )
{
        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
        strftime( szTimeCOutput , 30 , "%Y-%m-%d
%H:%M:%S.000", &when );

        return;
}
```

time.c

```
//          File:               TIME.C
//                                      Microsoft TPC-C Kit
Ver. 4.42
//                                      Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//          Purpose:    Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//=======================================
==========================
//
// Function name: TimeNow
//
//=======================================
==========================

long TimeNow()
{
  long                  time_now;
  struct      _timeb el_time;

#ifdef DEBUG
   printf("[%ld]DBG: Entering TimeNow()\n", (int) GetCurrentThreadId());
#endif

  _ftime(&el_time);

  time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

  return time_now;
}
```

strings.c

```
//          File:               STRINGS.C
//                                      Microsoft TPC-C Kit
Ver. 4.51
//                                      Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//          Purpose:    Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
```

```
#include <ctype.h>

//================================================
=========================
//
// Function name: MakeAddress
//
//================================================
=========================

void MakeAddress(char *street_1,
                                                char *street_2,
                                                char *city,
                                                char *state,
                                                char *zip)
{

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAddress()\n", (int)
GetCurrentThreadId());
#endif

   MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
   MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
   MakeAlphaString (10, 20, ADDRESS_LEN, city);
   MakeAlphaString ( 2,  2, STATE_LEN, state);
   MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
   printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s,
state: %s, zip: %s\n",
                                (int) GetCurrentThreadId(), street_1,
street_2, city, state, zip);
#endif

   return;

}


//================================================
=========================
//
// Function name: LastName
//
//================================================
=========================

void LastName(int  num,
                        char *name)
{
   static char *n[] =
   {                    "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                        "ESE" , "ANTI" , "CALLY", "ATION", "EING"
   };

#ifdef DEBUG
   printf("[%ld]DBG: Entering LastName()\n", (int) GetCurrentThreadId());
#endif

   if ((num >= 0) && (num < 1000))
   {
                        strcpy(name, n[(num/100)%10]);
                        strcat(name, n[(num/10)%10]);
                        strcat(name, n[(num/1)%10]);

                        if (strlen(name) < LAST_NAME_LEN)
                        {
                                PaddString(LAST_NAME_LEN,
name);
                        }

   }
   else
   {
                        printf("'\nError in LastName()... num <%ld> out
of range (0,999)\n", num);
                        exit(-1);
   }


#ifdef DEBUG
   printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                                (int) GetCurrentThreadId(), num,
num/100, (num/10)%10, num%10);
            printf("[%ld]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

   return;

}


//================================================
=========================
//
// Function name: MakeAlphaString
//
//================================================
=========================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//A string [x, y] represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x,
maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a
minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non
issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
                int                     len;
                int                     i;
   char    cc = 'a';
                static          char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
";
                static          int             chArrayMax = 61;

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAlphaString()\n", (int)
GetCurrentThreadId());
#endif

                len= RandomNumber(x, y);

                for (i=0; i<len; i++)
                        str[i] = chArray[RandomNumber(0,chArrayMax)];
                str[len] = 0;

   return len;
}

int MakeAlphaStringPadded( int  minLen, int  maxLen, int  padLen, char *str)
{
                int                     len;
                int                     i;
   char    cc = 'a';
                static          char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
";
                static          int             chArrayMax = 61;

#ifdef DEBUG
   printf("[%ld]DBG: Entering MakeAlphaStringPadded()\n", (int)
GetCurrentThreadId());
#endif

                len= RandomNumber(minLen, maxLen);

                for (i=0; i<len; i++)
```

```
                        str[i] = chArray[RandomNumber(0,chArrayMax)];
                if (len < padLen)
                        memset(str+len, ' ', padLen - len);
                str[padLen] = 0;
    return padLen;
}



//=============================================
========================
//
// Function name: MakeOriginalAlphaString
//
//=============================================
========================

int MakeOriginalAlphaString(int  x,

                int  y,

                        int  z,

                char *str,

                int  percent)
{
    int                 len;
    int                 val;
    int                 start;

#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify prercentage is valid
    if ((percent < 0) || (percent > 100))
    {
                        printf("MakeOrigianlAlphaString: Invalid
percentage: %d\n", percent);
                        exit(-1);
    }

    // verify string is at least 8 chars in length
    if (x < 8)
    {
                        printf("MakeOriginalAlphaString: string length
must be >= 8\n");
                        exit(-1);
    }

    len = MakeAlphaString(x,y, z, str);
    // Make Alpha String

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                        start = RandomNumber(0, len - 8);
                        strncpy(str + start, "ORIGINAL", 8);
    }


#ifdef DEBUG
    printf("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
                                        (int) GetCurrentThreadId(), str);
#endif

    return len;
}


//=============================================
========================
//
// Function name: MakeNumberString
//
//=============================================
========================
int MakeNumberString(int  x, int  y, int  z, char *str)
{
                char tmp[16];

                //MakeNumberString is always called MakeZipNumberString(16,
16, 16, string)

                memset(str, '0', 16);
                itoa(RandomNumber(0, 99999999), tmp, 10);
                memcpy(str, tmp, strlen(tmp));

                itoa(RandomNumber(0, 99999999), tmp, 10);
                memcpy(str+8, tmp, strlen(tmp));

                str[16] = 0;

    return 16;
}


//=============================================
========================
//
// Function name: MakeZipNumberString
//
//=============================================
========================
{
int MakeZipNumberString(int  x, int  y, int  z, char *str)
                char tmp[16];

                //MakeZipNumberString is always called
MakeZipNumberString(9, 9, 9, string)

                strcpy(str, "000011111");

                itoa(RandomNumber(0, 9999), tmp, 10);
                memcpy(str, tmp, strlen(tmp));

    return 9;
}


//=============================================
========================
//
// Function name: InitString
//
//=============================================
========================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

                memset(str, ' ', len);
    str[len] = 0;
}


//=============================================
========================
// Function name: InitAddress
//
// Description:
//
//=============================================
========================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char
*zip)
{
                memset(street_1, ' ', ADDRESS_LEN+1);
                memset(street_2, ' ', ADDRESS_LEN+1);
                memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;
```

```
		memset(state, ' ', STATE_LEN+1);
	state[STATE_LEN+1] = 0;

		memset(zip, ' ', ZIP_LEN+1);
	zip[ZIP_LEN+1] = 0;
}


//==============================================
========================
//
// Function name: PaddString
//
//==============================================
========================

void PaddString(int max, char *name)
{
		int				len;

		len = strlen(name);
		if ( len < max )
			memset(name+len, ' ', max - len);
		name[max] = 0;

		return;
}
```

random.c

```
//		File:				RANDOM.C
//					Microsoft TPC-C Kit
Ver. 4.42
//					Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//		Purpose:	Random number generation routines for database
loader


// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A		16807
#define Q		127773	/* M div A */
#define M		2147483647
#define R		2836	/* M mod A */
#define Thread		__declspec(thread)

// Globals
long		Thread Seed = 0;	/* thread local seed */

/*************************************************************
****************
*					*
* random -					*
*	Implements a GOOD pseudo random number generator.  This
generator	*
*	will/should? run the complete period before repeating.		*
*					*
* Copied from:					*
*	Random Numbers Generators: Good Ones Are Hard to Find.
*
*	Communications of the ACM - October 1988 Volume 31 Number 10
*
*					*
* Machine Dependencies:					*
*	long must be 2 ^ 31 - 1 or greater.					*
*					*
*************************************************************
***************/

/*************************************************************
****************
* seed - load the Seed value used in irand and drand.  Should be used before
*
*	first call to irand or drand.					*
*************************************************************
***************/

void seed(long val)
{

#ifdef DEBUG
	printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
		printf("Old Seed %ld New Seed %ld\n",Seed, val);
#endif

	if ( val < 0 )
		val = abs(val);

	Seed = val;

}


/*************************************************************
***************
* irand - returns a 32 bit integer pseudo random number with a period of
*
*	1 to 2 ^ 32 - 1.					*
*					*
* parameters:					*
*	none.					*
*					*
* returns:					*
*	32 bit integer - defined as long ( see above ).		*
*					*
* side effects:					*
*	seed get recomputed.					*
*************************************************************
**************/

long irand()
{
	register long	s;	/* copy of seed */
	register long	test;	/* test flag */
	register long	hi;	/* tmp value for speed */
	register long	lo;	/* tmp value for speed */

#ifdef DEBUG
	printf("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

	s = Seed;
	hi = s / Q;
	lo = s % Q;

	test = A * lo - R * hi;
	if ( test > 0 )
		Seed = test;
	else
		Seed = test + M;

	return( Seed );
}


/*************************************************************
***************
*					*
* drand - returns a double pseudo random number between 0.0 and 1.0.
*
*	See irand.					*
*************************************************************
**************/
double drand()
{

#ifdef DEBUG
	printf("[%ld]DBG: Entering drand()...\n", (int) GetCurrentThreadId());
```

```
#endif

    return( (double)irand() / 2147483647.0);
}


//=============================================
=========================
// Function   : RandomNumber
//
// Description:
//=============================================
=========================
long RandomNumber(long lower, long upper)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

                if ( upper == lower )        /* pgd 08-13-96 perf enhancement
*/
                        return lower;

                upper++;

    if ( upper <= lower )
                        rand_num = upper;
    else
                        rand_num = lower + irand() % (upper - lower);
/* pgd 08-13-96 perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                    (int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                                    long upper)
{   long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering RandomNumber()...\n", (int)
GetCurrentThreadId());
#endif

                upper++;

    if ((upper <= lower))
                        rand_num = upper;
    else
                        rand_num = lower + irand() % ((upper > lower)
? upper - lower : upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                                    (int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}
#endif

//=============================================
=========================
// Function   : NURand
//
// Description:
//=============================================
=========================
long NURand(int iConst,
                long x,
                long y,
                long C)
{
    long rand_num;

#ifdef DEBUG
    printf("[%ld]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

    rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) %
(y-x+1))+x;

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(),
rand_num);
#endif

} return rand_num;
```

getargs.c

```
//              File:                   GETARGS.C
//                                                  Microsoft TPC-C Kit
Ver. 4.42
//                                                  Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//              Purpose:        Source file for command line processing


// Includes
#include "tpcc.h"

//=============================================
=========================
//
// Function name: GetArgsLoader
//
//=============================================
=========================

void GetArgsLoader(int argc, char **argv, TPCCLDR_ARGS *pargs)
{
    int                 i;
    char        *ptr;

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoader()\n", (int)
GetCurrentThreadId());
#endif

    /* init args struct with some useful values */
    pargs->server                       =  SERVER;
    pargs->user                         =  USER;
    pargs->password                     =  PASSWORD;
    pargs->database                     =  DATABASE;
    pargs->batch                        =  BATCH;
    pargs->num_warehouses                       =  UNDEF;
            pargs->tables_all                           =  TRUE;
            pargs->table_item                           =
FALSE;
            pargs->table_warehouse              =  FALSE;
            pargs->table_customer               =  FALSE;
            pargs->table_orders                         =
FALSE;
            pargs->loader_res_file              =  LOADER_RES_FILE;
```

```
        pargs->log_path
        = LOADER_LOG_PATH;
        pargs->pack_size                = DEFLDPACKSIZE;
        pargs->starting_warehouse       =
DEF_STARTING_WAREHOUSE;
        pargs->build_index                      =
BUILD_INDEX;
        pargs->index_order                      =
INDEX_ORDER;
        pargs->index_script_path = INDEX_SCRIPT_PATH;
        pargs->scale_down                       =
SCALE_DOWN;
        pargs->sleep_milli                      =
SLEEPMILLI;
        pargs->retry_limit                      =
RETRYLIMIT;

    /* check for zero command line args */
    if ( argc == 1 )
                GetArgsLoaderUsage();

    for (i = 1; i < argc; ++i)
    {
                if (argv[i][0] != '-' && argv[i][0] != '/')
                {
                printf("\nUnrecognized command");
                GetArgsLoaderUsage();
                exit(1);
                }

                ptr = argv[i];

                switch (ptr[1])
                {
                case '?':    /* Fall throught */
                                GetArgsLoaderUsage();
                                break;

                case 'D':
                                pargs->database =
ptr+2;
                                break;

                case 'P':
                                pargs->password =
ptr+2;
                                break;

                case 'S':
                                pargs->server =
ptr+2;
                                break;

                case 'U':
                                pargs->user = ptr+2;
                                break;

                case 'b':
                                pargs->batch =
atol(ptr+2);
                                break;

                case 'W':
                                pargs-
>num_warehouses = atol(ptr+2);
                                break;

                case 's':
                                pargs-
>starting_warehouse = atol(ptr+2);
                                break;

                case 't':
                                {
                                pargs-
>tables_all = FALSE;
                                if
(strcmp(ptr+2,"item") == 0)
                                pargs->table_item = TRUE;
                                else if
(strcmp(ptr+2,"warehouse") == 0)
                                pargs->table_warehouse = TRUE;
                                else if
(strcmp(ptr+2,"customer") == 0)
                                pargs->table_customer = TRUE;
                                else if
(strcmp(ptr+2,"orders") == 0)
                                pargs->table_orders = TRUE;
                                else
                                {
                                printf("\nUnrecognized command");
                                GetArgsLoaderUsage();
                                exit(1);
                                }
                                break;
                }

                case 'f':
                                pargs->loader_res_file
= ptr+2;
                                break;

                case 'L':
                                pargs->log_path =
ptr+2;
                                break;

                case 'p':
                                pargs->pack_size =
atol(ptr+2);
                                break;

                case 'i':
                                pargs->build_index =
atol(ptr+2);
                                break;

                case 'o':
                                pargs->index_order =
atol(ptr+2);
                                break;

                case 'c':
                                pargs->scale_down =
atol(ptr+2);
                                break;

                case 'd':
                                pargs-
>index_script_path = ptr+2;
                                break;

                case 'R':
                                pargs->retry_limit =
atol(ptr+2);
                                break;

                case 'r':
                                pargs->sleep_milli =
atol(ptr+2);
                                break;

                default:
                GetArgsLoaderUsage();
                exit(-1);
                break;
        }
    }

    /* check for required args */
    if (pargs->num_warehouses == UNDEF )
    {
```

```
                printf("Number of Warehouses is required\n");
                exit(-2);
    }

    return;
}


//==============================================
=========================
//
// Function name: GetArgsLoaderUsage
//
//==============================================
=========================

void GetArgsLoaderUsage()
{

#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int)
GetCurrentThreadId());
#endif


    printf("TPCCLDR:\n\n");
        printf("Parameter                                   Default\n");
        printf("-------------------------------------------------------\n");
    printf("-W Number of Warehouses to Load             Required \n");
    printf("-S Server                                %s\n", SERVER);
    printf("-U Username                               %s\n", USER);
    printf("-P Password                               %s\n", PASSWORD);
    printf("-D Database                               %s\n", DATABASE);
        printf("-b Batch Size                             %ld\n", (long)
BATCH);
        printf("-p TDS packet size                        %ld\n",
(long) DEFLDPACKSIZE);
        printf("-f Loader Results Output Filename         %s\n",
LOADER_RES_FILE);
        printf("-s Starting Warehouse                     %ld\n",
(long) DEF_STARTING_WAREHOUSE);
        printf("-i Build Option (data = 0, data and index = 1)
%ld\n", (long) BUILD_INDEX);
        printf("-o Cluster Index Build Order (before = 1, after = 0)
%ld\n", (long) INDEX_ORDER);
        printf("-R Retry limit. # of times to retry open  %ld\n",
(long) RETRYLIMIT);
        printf("-r Retry sleep time in milliseconds.      %ld\n",
(long) SLEEPMILLI);
        printf("-c Build Scaled Database (normal = 0, tiny = 1)
%ld\n", (long) SCALE_DOWN);
        printf("-d Index Script Path                      %s\n",
INDEX_SCRIPT_PATH);
        printf("-t Table to Load                          all tables \n");

    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the '-t' parameter may be included multiple times to \n");
    printf("       specify multiple tables to be loaded \n");
    printf("     - 'item' loads ITEM table \n");
    printf("     - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables
\n");
    printf("     - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("     - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

        printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}
```

B.1 Database Options

DBOPT1.SQL

```
----------------------------------------------------------------
--                                                            --
--  File:   DBOPT1.SQL                                        --
--          Microsoft TPC-C Benchmark Kit Ver. 4.60           --
--          Copyright Microsoft, 2005                         --
--                                                            --
--          Sets database options for load                    --
--                                                            --
----------------------------------------------------------------
USE master
GO

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
GO

EXEC sp_dboption tpcc,'trunc. log on chkpt.',TRUE
GO

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION OFF
GO
ALTER DATABASE tpcc SET PAGE_VERIFY NONE
GO

USE tpcc
GO

CHECKPOINT
GO
```

DBOPT2.SQL

```
----------------------------------------------------------------
--                                                            --
--  File:   DBOPT2.SQL                                        --
--          Microsoft TPC-C Benchmark Kit Ver. 4.60           --
--          Copyright Microsoft, 2005                         --
--                                                            --
--          Sets database options after load                  --
--                                                            --
----------------------------------------------------------------
ALTER DATABASE tpcc SET RECOVERY FULL
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE     @msg                        varchar(50)

------------------------------------------------
--        OPTIONS FOR SQL SERVER 2000         --
-- Set option values for user-defined indexes --
------------------------------------------------

SET     @msg    = ' '
PRINT   @msg
```

```
SET     @msg    = 'Setting SQL Server indexoptions'
PRINT   @msg
SET     @msg    = ' '
PRINT   @msg

EXEC sp_indexoption         'customer',     'DisAllowPageLocks',        TRUE
EXEC sp_indexoption         'district',         'DisAllowPageLocks',    TRUE
EXEC sp_indexoption         'warehouse',    'DisAllowPageLocks',        TRUE
EXEC sp_indexoption         'stock',        'DisAllowPageLocks',        TRUE
EXEC sp_indexoption         'order_line',   'DisAllowRowLocks',     TRUE
EXEC sp_indexoption         'orders',       'DisAllowRowLocks',     TRUE
EXEC sp_indexoption         'new_order',    'DisAllowRowLocks',     TRUE
EXEC sp_indexoption         'item',         'DisAllowRowLocks',     TRUE
EXEC sp_indexoption         'item',         'DisAllowPageLocks',    FALSE
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '            Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT  name,
        lockflags
FROM    sysindexes
WHERE   object_id('warehouse')  = id OR
        object_id('district')   = id OR
        object_id('customer')   = id OR
        object_id('stock')      = id OR
        object_id('orders')     = id OR
        object_id('order_line') = id OR
        object_id('history')    = id OR
        object_id('new_order')  = id OR
        object_id('item')       = id
ORDER   BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,    'auto update statistics',     FALSE
EXEC sp_dboption tpcc,    'auto create statistics',     FALSE
GO

DECLARE @db_id  int,
        @tbl_id int

SET     @db_id  = DB_ID('tpcc')
SET     @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)

SET     @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)

SET     @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)

SET     @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO
```

B.2 Table definitions

Createdatabase.sql

```
--
-- HPTPCC Setup Kit
-- Create Database
--

-- Create main database files
--
-- Files are loaded into the load filegroup
-- Then Create[table]Indexes.sql moves them to the final filegroup

if exists (select * from sysdatabases where name='tpcc') drop database tpcc

CREATE DATABASE tpcc
ON PRIMARY
(               NAME                    = tpccRoot,
                FILENAME    = "g:\tpcc112000.mdf",
                SIZE                    = 12MB,
                FILEGROWTH              = 0),

FILEGROUP  cs_fg
                (NAME=cs1, FILENAME="g:\mnt\cs\1\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs2, FILENAME="g:\mnt\cs\2\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs3, FILENAME="g:\mnt\cs\3\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs4, FILENAME="g:\mnt\cs\4\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs5, FILENAME="g:\mnt\cs\5\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs6, FILENAME="g:\mnt\cs\6\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs7, FILENAME="g:\mnt\cs\7\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs8, FILENAME="g:\mnt\cs\8\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs9, FILENAME="g:\mnt\cs\9\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs10, FILENAME="g:\mnt\cs\10\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs11, FILENAME="g:\mnt\cs\11\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs12, FILENAME="g:\mnt\cs\12\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs13, FILENAME="g:\mnt\cs\13\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs14, FILENAME="g:\mnt\cs\14\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs15, FILENAME="g:\mnt\cs\15\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs16, FILENAME="g:\mnt\cs\16\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs17, FILENAME="g:\mnt\cs\17\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs18, FILENAME="g:\mnt\cs\18\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs19, FILENAME="g:\mnt\cs\19\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs20, FILENAME="g:\mnt\cs\20\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs21, FILENAME="g:\mnt\cs\21\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs22, FILENAME="g:\mnt\cs\22\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs23, FILENAME="g:\mnt\cs\23\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs24, FILENAME="g:\mnt\cs\24\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs25, FILENAME="g:\mnt\cs\25\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs26, FILENAME="g:\mnt\cs\26\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs27, FILENAME="g:\mnt\cs\27\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs28, FILENAME="g:\mnt\cs\28\",
SIZE=113083MB,FILEGROWTH=0),
                (NAME=cs29, FILENAME="g:\mnt\cs\29\",
SIZE=113083MB,FILEGROWTH=0),
```

```
        (NAME=cs30, FILENAME="g:\mnt\cs\30\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs31, FILENAME="g:\mnt\cs\31\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs32, FILENAME="g:\mnt\cs\32\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs33, FILENAME="g:\mnt\cs\33\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs34, FILENAME="g:\mnt\cs\34\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs35, FILENAME="g:\mnt\cs\35\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs36, FILENAME="g:\mnt\cs\36\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs37, FILENAME="g:\mnt\cs\37\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs38, FILENAME="g:\mnt\cs\38\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs39, FILENAME="g:\mnt\cs\39\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs40, FILENAME="g:\mnt\cs\40\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs41, FILENAME="g:\mnt\cs\41\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs42, FILENAME="g:\mnt\cs\42\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs43, FILENAME="g:\mnt\cs\43\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs44, FILENAME="g:\mnt\cs\44\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs45, FILENAME="g:\mnt\cs\45\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs46, FILENAME="g:\mnt\cs\46\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs47, FILENAME="g:\mnt\cs\47\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs48, FILENAME="g:\mnt\cs\48\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs49, FILENAME="g:\mnt\cs\49\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs50, FILENAME="g:\mnt\cs\50\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs51, FILENAME="g:\mnt\cs\51\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs52, FILENAME="g:\mnt\cs\52\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs53, FILENAME="g:\mnt\cs\53\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs54, FILENAME="g:\mnt\cs\54\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs55, FILENAME="g:\mnt\cs\55\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs56, FILENAME="g:\mnt\cs\56\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs57, FILENAME="g:\mnt\cs\57\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs58, FILENAME="g:\mnt\cs\58\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs59, FILENAME="g:\mnt\cs\59\",
SIZE=113083MB,FILEGROWTH=0),
        (NAME=cs60, FILENAME="g:\mnt\cs\60\",
SIZE=113083MB,FILEGROWTH=0),


FILEGROUP   misc_fg


        (NAME=misc1, FILENAME="g:\mnt\misc\1\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc2, FILENAME="g:\mnt\misc\2\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc3, FILENAME="g:\mnt\misc\3\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc4, FILENAME="g:\mnt\misc\4\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc5, FILENAME="g:\mnt\misc\5\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc6, FILENAME="g:\mnt\misc\6\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc7, FILENAME="g:\mnt\misc\7\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc8, FILENAME="g:\mnt\misc\8\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc9, FILENAME="g:\mnt\misc\9\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc10, FILENAME="g:\mnt\misc\10\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc11, FILENAME="g:\mnt\misc\11\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc12, FILENAME="g:\mnt\misc\12\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc13, FILENAME="g:\mnt\misc\13\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc14, FILENAME="g:\mnt\misc\14\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc15, FILENAME="g:\mnt\misc\15\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc16, FILENAME="g:\mnt\misc\16\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc17, FILENAME="g:\mnt\misc\17\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc18, FILENAME="g:\mnt\misc\18\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc19, FILENAME="g:\mnt\misc\19\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc20, FILENAME="g:\mnt\misc\20\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc21, FILENAME="g:\mnt\misc\21\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc22, FILENAME="g:\mnt\misc\22\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc23, FILENAME="g:\mnt\misc\23\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc24, FILENAME="g:\mnt\misc\24\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc25, FILENAME="g:\mnt\misc\25\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc26, FILENAME="g:\mnt\misc\26\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc27, FILENAME="g:\mnt\misc\27\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc28, FILENAME="g:\mnt\misc\28\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc29, FILENAME="g:\mnt\misc\29\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc30, FILENAME="g:\mnt\misc\30\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc31, FILENAME="g:\mnt\misc\31\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc32, FILENAME="g:\mnt\misc\32\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc33, FILENAME="g:\mnt\misc\33\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc34, FILENAME="g:\mnt\misc\34\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc35, FILENAME="g:\mnt\misc\35\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc36, FILENAME="g:\mnt\misc\36\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc37, FILENAME="g:\mnt\misc\37\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc38, FILENAME="g:\mnt\misc\38\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc39, FILENAME="g:\mnt\misc\39\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc40, FILENAME="g:\mnt\misc\40\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc41, FILENAME="g:\mnt\misc\41\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc42, FILENAME="g:\mnt\misc\42\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc43, FILENAME="g:\mnt\misc\43\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc44, FILENAME="g:\mnt\misc\44\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc45, FILENAME="g:\mnt\misc\45\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc46, FILENAME="g:\mnt\misc\46\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc47, FILENAME="g:\mnt\misc\47\",
SIZE=62672MB,FILEGROWTH=0),
```

```
        (NAME=misc48, FILENAME="g:\mnt\misc\48\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc49, FILENAME="g:\mnt\misc\49\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc50, FILENAME="g:\mnt\misc\50\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc51, FILENAME="g:\mnt\misc\51\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc52, FILENAME="g:\mnt\misc\52\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc53, FILENAME="g:\mnt\misc\53\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc54, FILENAME="g:\mnt\misc\54\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc55, FILENAME="g:\mnt\misc\55\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc56, FILENAME="g:\mnt\misc\56\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc57, FILENAME="g:\mnt\misc\57\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc58, FILENAME="g:\mnt\misc\58\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc59, FILENAME="g:\mnt\misc\59\",
SIZE=62672MB,FILEGROWTH=0),
        (NAME=misc60, FILENAME="g:\mnt\misc\60\",
SIZE=62672MB,FILEGROWTH=0),



FILEGROUP   load_fg
        (NAME=load1, FILENAME="g:\mnt\load\1\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load2, FILENAME="g:\mnt\load\2\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load3, FILENAME="g:\mnt\load\3\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load4, FILENAME="g:\mnt\load\4\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load5, FILENAME="g:\mnt\load\5\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load6, FILENAME="g:\mnt\load\6\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load7, FILENAME="g:\mnt\load\7\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load8, FILENAME="g:\mnt\load\8\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load9, FILENAME="g:\mnt\load\9\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load10, FILENAME="g:\mnt\load\10\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load11, FILENAME="g:\mnt\load\11\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load12, FILENAME="g:\mnt\load\12\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load13, FILENAME="g:\mnt\load\13\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load14, FILENAME="g:\mnt\load\14\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load15, FILENAME="g:\mnt\load\15\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load16, FILENAME="g:\mnt\load\16\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load17, FILENAME="g:\mnt\load\17\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load18, FILENAME="g:\mnt\load\18\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load19, FILENAME="g:\mnt\load\19\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load20, FILENAME="g:\mnt\load\20\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load21, FILENAME="g:\mnt\load\21\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load22, FILENAME="g:\mnt\load\22\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load23, FILENAME="g:\mnt\load\23\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load24, FILENAME="g:\mnt\load\24\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load25, FILENAME="g:\mnt\load\25\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load26, FILENAME="g:\mnt\load\26\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load27, FILENAME="g:\mnt\load\27\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load28, FILENAME="g:\mnt\load\28\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load29, FILENAME="g:\mnt\load\29\",
SIZE=129000MB,FILEGROWTH=0),
        (NAME=load30, FILENAME="g:\mnt\load\30\",
SIZE=129000MB,FILEGROWTH=0)

  LOG ON
(       NAME                    = tpcc_log,
        FILENAME    = "L:",
        SIZE                    = 150000MB,
        FILEGROWTH              = 0)
```

Createtables.sql

```
----------------------------------------------------------------
== File:  TABLES.SQL                            --      --
--      Microsoft TPC-C Benchmark Kit Ver. 4.51         --
--      Copyright Microsoft, 2003                       --
--                                              --
--      Creates TPC-C tables                    --
----------------------------------------------------------------

--  May 13, 2005 - CJL
--  Modified from Shiloh kit.  Reorder columns; change numeric to money.
--

SET ANSI_NULL_DFLT_OFF ON
go

use tpcc
go

-------------------------------------
-- Remove all existing TPC-C tables
-------------------------------------
if exists ( select name from sysobjects where name = 'warehouse' )
        drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
        drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
        drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
        drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
        drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
        drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
        drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
        drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
        drop table stock
go

--------------------
-- Create new tables
--------------------
create table warehouse
(
```

```
        w_id                    int,
        w_ytd                   money,
        w_tax
        smallmoney,
        w_name                  char(10),
        w_street_1              char(20),
        w_street_2              char(20),
        w_city                  char(20),
        w_state                 char(2),
        w_zip                   char(9)
) on load_fg
go

create table district
(
        d_id                    tinyint,
        d_w_id                  int,
        d_ytd                   money,

        d_next_o_id             int,

        d_tax
        smallmoney,
        d_name                  char(10),
        d_street_1              char(20),
        d_street_2              char(20),
        d_city                  char(20),
        d_state                 char(2),
        d_zip                   char(9)
) on load_fg
go

create table customer
(
        c_id                    int,
        c_d_id                  tinyint,
        c_w_id                  int,
        c_discount              smallmoney,
        c_credit_lim    money,
        c_last                  char(16),

        c_first                 char(16),
        c_credit                char(2),
        c_balance               money,

        c_ytd_payment           money,

        c_payment_cnt           smallint,
        c_delivery_cnt          smallint,
        c_street_1              char(20),
        c_street_2              char(20),
        c_city                  char(20),
        c_state                 char(2),
        c_zip                   char(9),
        c_phone                 char(16),
        c_since                 datetime,
        c_middle        char(2),
        c_data                  char(500)
) on load_fg
go



-- Use the following table option if using c_data varchar(max)
-- sp_tableoption 'customer','large value types out of row','1'
-- go

create table history
(
        h_c_id                  int,
        h_c_d_id        tinyint,
        h_c_w_id        int,
        h_d_id                  tinyint,
        h_w_id                  int,
        h_date                  datetime,
        h_amount        smallmoney,
        h_data                  char(24)
) on load_fg
go



create table new_order
(
        no_o_id                 int,
        no_d_id                 tinyint,
        no_w_id                 int
) on load_fg
go



create table orders
(
        o_id                    int,
        o_d_id                  tinyint,
        o_w_id                  int,
        o_c_id                  int,
        o_carrier_id    tinyint,
        o_ol_cnt                tinyint,
        o_all_local             tinyint,
        o_entry_d               datetime
) on load_fg
go

create table order_line
(
        ol_o_id                 int,
        ol_d_id                 tinyint,
        ol_w_id                 int,
        ol_number               tinyint,
        ol_i_id                 int,
        ol_delivery_d   datetime,
        ol_amount               smallmoney,
        ol_supply_w_id          int,
        ol_quantity             smallint,
        ol_dist_info    char(24)
) on load_fg
go

create table item
(
        i_id                    int,
        i_name                  char(24),

        i_price
        smallmoney,
        i_data                  char(50),
        i_im_id                 int
) on load_fg
go

create table stock
(
        s_i_id                  int,
        s_w_id                  int,
        s_quantity              smallint,
        s_ytd                   int,

        s_order_cnt             smallint,
        s_remote_cnt            smallint,
        s_data                  char(50),

        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24)
) on load_fg
go
```

newcreateCustomerindexs.sql

```
--
-- HP TPCC Setup
-- Create Indexes for Customer
-- and move the table from LoadFg to CustomerStockFg
--
use tpcc
go

if exists (select name from sysindexes where name ='customer_c1')
  drop index customer.customer_c1

if exists (select name from sysindexes where name ='history_c1')
  drop index history.history_c1
go


create unique clustered index customer_c1 on customer(c_w_id, c_id,
c_d_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go

create unique nonclustered index customer_nc1 on customer(c_w_id, c_d_id,
c_last, c_first, c_id)
with SORT_IN_TEMPDB, fillfactor=100
go

create clustered index history_c1 on history(h_c_w_id, h_date, h_c_d_id,
h_c_id, h_amount)
 with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

newcreateItemindexes.sql

```
-- HPTPCC Setup Kit
-- Create Item Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go

create unique clustered index item_c1 on item(i_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

newcreateOrdersindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create ORDERS Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc


go

if exists ( select name from sysindexes where name = 'orders_c1' )
                drop index orders.orders_c1

if exists ( select name from sysindexes where name = 'order_line_c1' )
                drop index order_line.order_line_c1

if exists ( select name from sysindexes where name = 'new_order_c1' )
                drop index new_order.new_order_c1
go

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

--create index orders_nc1 on orders(o_w_id, o_d_id, o_c_id, o_id)
--with SORT_IN_TEMPDB, fillfactor=100
--go

create unique clustered index new_order_c1 on new_order(no_w_id,
no_d_id, no_o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index order_line_c1 on order_line(ol_w_id, ol_d_id,
ol_o_id, ol_number)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go
```

newcreateStockindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create STOCK Indexes
-- and move the table from LoadFg to MiscFg
--

use tpcc
go

create unique clustered index warehouse_c1 on warehouse(w_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100, SORT_IN_TEMPDB on misc_fg
go

create unique clustered index stock_c1 on stock(s_i_id,s_w_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go
```

newcreateWarehouseindexes.sql

```
--
-- HPTPCC Setup Kit
-- Create WAREHOUSE Indexes
-- and move the table from LoadFg to MiscFg
--

use tpcc
go

if exists (select name from sysindexes where name ='stock_c1')
  drop index stock.stock_c1
go


create unique clustered index warehouse_c1 on warehouse(w_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index district_c1 on district(d_w_id, d_id)
with fillfactor=100, SORT_IN_TEMPDB on misc_fg
go
```

```
create unique clustered index stock_c1 on stock(s_i_id,s_w_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go
```

B.3 Stored Procedures

Create_NewOrd.sql

```
------------------------------------------------------------------
--                                                              --
-- File:   NEWORD.SQL                                           --
--         Microsoft TPC-C Benchmark Kit Ver. 4.62              --
--         Copyright Microsoft, 2005                            --
--                                                              --
--         Creates neworder stored procedure                    --
--                                                              --
--         Interface Level:    4.20.000                         --
--                                                              --
------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder'
)
   DROP PROCEDURE tpcc_neworder
GO

CREATE PROCEDURE    tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0


AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @commit_flag    = 0,
            @li_no          = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

-----------------------
--  process orderlines
-----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

              @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

              @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
```

```
                    WHEN 3 THEN @ol_qty3
                    WHEN 4 THEN @ol_qty4
                    WHEN 5 THEN @ol_qty5
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
    SELECT  @i_price    = i_price,
            @i_name     = i_name,
            @i_data     = i_data
    FROM    item WITH (repeatableread)
    WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
    UPDATE  stock
    SET     s_ytd           = s_ytd + @li_qty,
            @s_quantity     = s_quantity   = s_quantity - @li_qty +
                        CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
            s_order_cnt     = s_order_cnt + 1,
            s_remote_cnt    = s_remote_cnt +
                        CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
            @s_data         = s_data,
            @s_dist         = CASE @d_id
                        WHEN 1  THEN s_dist_01
                        WHEN 2  THEN s_dist_02
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-- if there actually is a stock (and item) with these ids, go to work
-----------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                    @d_id,
                                    @w_id,
                                    @li_no,
                                    @li_id,
                                    'dec 31, 1899',
                                    @i_price * @li_qty,
                                    @li_s_w_id,
                                    @li_qty,
                                    @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
            SELECT  ",0,",0,0
            SELECT  @commit_flag    = 0
                                END
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                            @d_id,
                            @w_id,
                            @c_id_local,
                            0,
                            @o_ol_cnt,
                            @o_all_local,
                            @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                @d_id,
                                @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
---------------------------------
--  all that work for nuthin!!!
---------------------------------
        ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO

-- Create SP0
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp0tpcc_neworder' )
    DROP PROCEDURE sp0tpcc_neworder
GO
```

```
CREATE PROC sp0tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0


AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n


-----------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

               @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

------------------------
--  update stock values
------------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
```

```
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
    IF (@@rowcount > 0)
    BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
                END
        SELECT  @commit_flag    = 0
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                    COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @c_last,
            @c_id,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP0


-- Create SP1
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp1tpcc_neworder' )
    DROP PROCEDURE sp1tpcc_neworder
GO

CREATE PROC sp1tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
```

```
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
        SELECT  @li_id = CASE @li_no
                            WHEN 1 THEN @i_id1
                            WHEN 2 THEN @i_id2
                            WHEN 3 THEN @i_id3
                            WHEN 4 THEN @i_id4
                            WHEN 5 THEN @i_id5
                            WHEN 6 THEN @i_id6
                            WHEN 7 THEN @i_id7
                            WHEN 8 THEN @i_id8
                            WHEN 9 THEN @i_id9
                            WHEN 10 THEN @i_id10
                            WHEN 11 THEN @i_id11
                            WHEN 12 THEN @i_id12
                            WHEN 13 THEN @i_id13
                            WHEN 14 THEN @i_id14
                            WHEN 15 THEN @i_id15
                        END,

        @li_s_w_id = CASE @li_no
                            WHEN 1 THEN @s_w_id1
                            WHEN 2 THEN @s_w_id2
                            WHEN 3 THEN @s_w_id3
                            WHEN 4 THEN @s_w_id4
                            WHEN 5 THEN @s_w_id5
                            WHEN 6 THEN @s_w_id6
                            WHEN 7 THEN @s_w_id7
                            WHEN 8 THEN @s_w_id8
                            WHEN 9 THEN @s_w_id9
                            WHEN 10 THEN @s_w_id10
                            WHEN 11 THEN @s_w_id11
                            WHEN 12 THEN @s_w_id12
                            WHEN 13 THEN @s_w_id13
                            WHEN 14 THEN @s_w_id14
                            WHEN 15 THEN @s_w_id15
                        END,

        @li_qty = CASE  @li_no
                            WHEN 1 THEN @ol_qty1
                            WHEN 2 THEN @ol_qty2
                            WHEN 3 THEN @ol_qty3
                            WHEN 4 THEN @ol_qty4
                            WHEN 5 THEN @ol_qty5
                            WHEN 6 THEN @ol_qty6
                            WHEN 7 THEN @ol_qty7
                            WHEN 8 THEN @ol_qty8
                            WHEN 9 THEN @ol_qty9
                            WHEN 10 THEN @ol_qty10
                            WHEN 11 THEN @ol_qty11
                            WHEN 12 THEN @ol_qty12
                            WHEN 13 THEN @ol_qty13
                            WHEN 14 THEN @ol_qty14
                            WHEN 15 THEN @ol_qty15
                        END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
                                  WHEN 3  THEN s_dist_03
                                  WHEN 4  THEN s_dist_04
                                  WHEN 5  THEN s_dist_05
                                  WHEN 6  THEN s_dist_06
                                  WHEN 7  THEN s_dist_07
                                  WHEN 8  THEN s_dist_08
                                  WHEN 9  THEN s_dist_09
                                  WHEN 10 THEN s_dist_10
                                  END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-----------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-----------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
```

```
                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                                END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                           @d_id,
                           @w_id)

------------------------
--  select warehouse tax
------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP1



-- Create SP2
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp2tpcc_neworder' )
   DROP PROCEDURE sp2tpcc_neworder
GO

CREATE PROC sp2tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
        @o_id           = d_next_o_id,
        d_next_o_id     = d_next_o_id + 1,
        @o_entry_d      = GETDATE(),
        @li_no          = 0,
        @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
        d_id            = @d_id

-----------------------
--  process orderlines
-----------------------
```

```
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
                       WHEN 6 THEN @ol_qty6
                       WHEN 7 THEN @ol_qty7
                       WHEN 8 THEN @ol_qty8
                       WHEN 9 THEN @ol_qty9
                       WHEN 10 THEN @ol_qty10
                       WHEN 11 THEN @ol_qty11
                       WHEN 12 THEN @ol_qty12
                       WHEN 13 THEN @ol_qty13
                       WHEN 14 THEN @ol_qty14
                       WHEN 15 THEN @ol_qty15
                     END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

----------------------
--  update stock values
----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
      INSERT INTO order_line VALUES(  @o_id,
                                      @d_id,
                                      @w_id,
                                      @li_no,
                                      @li_id,
                                      'dec 31, 1899',
                                      @i_price * @li_qty,
                                      @li_s_w_id,
                                      @li_qty,
                                      @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
      SELECT  @i_name,
              @s_quantity,
              b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                   (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
              @i_price,
              @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
      SELECT  ",0,",0,0
      SELECT  @commit_flag    = 0
                                END
           END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                               @d_id,
                               @w_id,
                               @c_id_local,
```

```
                    0,
                    @o_ol_cnt,
                    @o_all_local,
                    @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                           @d_id,
                           @w_id)

-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                           COMMIT TRANSACTION n
   ELSE
---------------------------------
--  all that work for nuthin!!!
---------------------------------
      ROLLBACK TRANSACTION n

-------------------------------
--  return order data to client
-------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP2



-- Create SP3
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp3tpcc_neworder' )
   DROP PROCEDURE sp3tpcc_neworder
GO

CREATE PROC sp3tpcc_neworder
                   @w_id           int,
                   @d_id           int,
                   @o_id           int,
                   @o_ol_cnt       tinyint,
                   @o_all_local    tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                   @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                   @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                   @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                   @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                   @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                   @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                   @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                   @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                   @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                   @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-- get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                    END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
```

```
                    WHEN 12 THEN @s_w_id12
                    WHEN 13 THEN @s_w_id13
                    WHEN 14 THEN @s_w_id14
                    WHEN 15 THEN @s_w_id15
                  END,

        @li_qty = CASE  @li_no
                    WHEN 1 THEN @ol_qty1
                    WHEN 2 THEN @ol_qty2
                    WHEN 3 THEN @ol_qty3
                    WHEN 4 THEN @ol_qty4
                    WHEN 5 THEN @ol_qty5
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                  END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                              CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                              CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                              WHEN 1  THEN s_dist_01
                              WHEN 2  THEN s_dist_02
                              WHEN 3  THEN s_dist_03
                              WHEN 4  THEN s_dist_04
                              WHEN 5  THEN s_dist_05
                              WHEN 6  THEN s_dist_06
                              WHEN 7  THEN s_dist_07
                              WHEN 8  THEN s_dist_08
                              WHEN 9  THEN s_dist_09
                              WHEN 10 THEN s_dist_10
                            END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                     (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  ",0,",0,0
        SELECT  @commit_flag    = 0
                                END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating

---  SELECT  @c_last     = c_last,----------
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                            @d_id,
                            @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                            COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
```

```
        @commit_flag
END
GO
-- End SP3



-- Create SP4
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp4tpcc_neworder' )
   DROP PROCEDURE sp4tpcc_neworder
GO

CREATE PROC sp4tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @s_quantity     smallint,
        @remote_flag    int,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                      END,

               @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                      END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
       UPDATE  stock
```

```
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                               END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                       (patindex('%ORIGINAL%',@s_data) > 0) )
THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
              SELECT  '',0,'',0,0
              SELECT  @commit_flag    = 0
                                  END
                  END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
      SELECT  @c_last     = c_last,
              @c_discount = c_discount,
              @c_credit   = c_credit,
              @c_id_local = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
      INSERT INTO orders VALUES ( @o_id,
                                  @d_id,
                                  @w_id,
                                  @c_id_local,
                                  0,
                                  @o_ol_cnt,
                                  @o_all_local,
                                  @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
      INSERT INTO new_order VALUES (  @o_id,
                                      @d_id,
                                      @w_id)

------------------------
--  select warehouse tax
------------------------
      SELECT  @w_tax  = w_tax
      FROM    warehouse WITH (repeatableread)
      WHERE   w_id    = @w_id

      IF (@commit_flag = 1)
          COMMIT TRANSACTION n
      ELSE
          ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
      SELECT  @w_tax,
              @d_tax,
              @o_id,
              @c_last,
              @c_discount,
              @c_credit,
              @o_entry_d,
              @commit_flag
END
GO
-- End SP4



-- Create SP5
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp5tpcc_neworder' )
    DROP PROCEDURE sp5tpcc_neworder
GO

CREATE PROC sp5tpcc_neworder
                  @w_id           int,
                  @d_id           tinyint,
                  @c_id           int,
                  @o_ol_cnt       tinyint,
                  @o_all_local    tinyint,
                  @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                  @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                  @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                  @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                  @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                  @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                  @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                  @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                  @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                  @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                  @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                  @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                  @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
```

```
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN

        SELECT  @li_no  = @li_no + 1
-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

                @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

                @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

----------------------
--  update stock values
----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-----------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
```

```
                        @li_id,
                        'dec 31, 1899',
                        @i_price * @li_qty,
                        @li_s_w_id,
                        @li_qty,
                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                        END
                        ELSE
                        BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                        END
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local)

-------------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                        @d_id,
                        @w_id)

-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP5



-- Create SP6
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp6tpcc_neworder' )
   DROP PROCEDURE sp6tpcc_neworder
GO

CREATE PROC sp6tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @o_entry_d      datetime,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
```

```
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
                       WHEN 6 THEN @ol_qty6
                       WHEN 7 THEN @ol_qty7
                       WHEN 8 THEN @ol_qty8
                       WHEN 9 THEN @ol_qty9
                       WHEN 10 THEN @ol_qty10
                       WHEN 11 THEN @ol_qty11
                       WHEN 12 THEN @ol_qty12
                       WHEN 13 THEN @ol_qty13
                       WHEN 14 THEN @ol_qty14
                       WHEN 15 THEN @ol_qty15
                     END

---------------------------------------
--  get item data (no one updates item)
---------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                      (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                                 END
                                 ELSE
                                 BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
         SELECT  '',0,'',0,0
         SELECT  @commit_flag    = 0
                                 END
              END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
```

```
        @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                            @d_id,
                            @w_id,
                            @c_id_local,
                            0,
                            @o_ol_cnt,
                            @o_all_local,
                            @o_entry_d)

------------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                @d_id,
                                @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END

GO

End SP6



-- Create SP7
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp7tpcc_neworder' )
    DROP PROCEDURE sp7tpcc_neworder
GO

CREATE PROC sp7tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @o_id           int,
        @o_no           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

-----------------------
--  process orderlines
-----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
        SELECT  @li_id = CASE @li_no
                            WHEN 1 THEN @i_id1
                            WHEN 2 THEN @i_id2
                            WHEN 3 THEN @i_id3
                            WHEN 4 THEN @i_id4
                            WHEN 5 THEN @i_id5
                            WHEN 6 THEN @i_id6
                            WHEN 7 THEN @i_id7
                            WHEN 8 THEN @i_id8
                            WHEN 9 THEN @i_id9
                            WHEN 10 THEN @i_id10
                            WHEN 11 THEN @i_id11
                            WHEN 12 THEN @i_id12
                            WHEN 13 THEN @i_id13
                            WHEN 14 THEN @i_id14
                            WHEN 15 THEN @i_id15
                        END,
```

```
        @li_s_w_id = CASE @li_no
                     WHEN 1 THEN @s_w_id1
                     WHEN 2 THEN @s_w_id2
                     WHEN 3 THEN @s_w_id3
                     WHEN 4 THEN @s_w_id4
                     WHEN 5 THEN @s_w_id5
                     WHEN 6 THEN @s_w_id6
                     WHEN 7 THEN @s_w_id7
                     WHEN 8 THEN @s_w_id8
                     WHEN 9 THEN @s_w_id9
                     WHEN 10 THEN @s_w_id10
                     WHEN 11 THEN @s_w_id11
                     WHEN 12 THEN @s_w_id12
                     WHEN 13 THEN @s_w_id13
                     WHEN 14 THEN @s_w_id14
                     WHEN 15 THEN @s_w_id15
                   END,

        @li_qty = CASE  @li_no
                     WHEN 1 THEN @ol_qty1
                     WHEN 2 THEN @ol_qty2
                     WHEN 3 THEN @ol_qty3
                     WHEN 4 THEN @ol_qty4
                     WHEN 5 THEN @ol_qty5
                     WHEN 6 THEN @ol_qty6
                     WHEN 7 THEN @ol_qty7
                     WHEN 8 THEN @ol_qty8
                     WHEN 9 THEN @ol_qty9
                     WHEN 10 THEN @ol_qty10
                     WHEN 11 THEN @ol_qty11
                     WHEN 12 THEN @ol_qty12
                     WHEN 13 THEN @ol_qty13
                     WHEN 14 THEN @ol_qty14
                     WHEN 15 THEN @ol_qty15
                   END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

------------------------
--  update stock values
------------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
------------------------------------------------------------
--  insert order_line data (using data from item and stock)
------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                     (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
      END
      ELSE
      BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
        SELECT  ",0,",0,0
        SELECT  @commit_flag    = 0
                END
            END
-------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
---------------------------------
--  all that work for nuthin!!!
```

```
---------------------------------
       ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP7



-- Create SP8
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp8tpcc_neworder' )
   DROP PROCEDURE sp8tpcc_neworder
GO

CREATE PROC sp8tpcc_neworder
                   @w_id           int,
                   @d_id           tinyint,
                   @c_id           int,
                   @o_ol_cnt       tinyint,
                   @o_all_local    tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                   @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                   @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                   @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                   @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                   @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                   @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                   @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                   @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                   @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                   @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
SELECT  @li_id = CASE @li_no
                   WHEN 1 THEN @i_id1
                   WHEN 2 THEN @i_id2
                   WHEN 3 THEN @i_id3
                   WHEN 4 THEN @i_id4
                   WHEN 5 THEN @i_id5
                   WHEN 6 THEN @i_id6
                   WHEN 7 THEN @i_id7
                   WHEN 8 THEN @i_id8
                   WHEN 9 THEN @i_id9
                   WHEN 10 THEN @i_id10
                   WHEN 11 THEN @i_id11
                   WHEN 12 THEN @i_id12
                   WHEN 13 THEN @i_id13
                   WHEN 14 THEN @i_id14
                   WHEN 15 THEN @i_id15
                 END,

       @li_s_w_id = CASE @li_no
                   WHEN 1 THEN @s_w_id1
                   WHEN 2 THEN @s_w_id2
                   WHEN 3 THEN @s_w_id3
                   WHEN 4 THEN @s_w_id4
                   WHEN 5 THEN @s_w_id5
                   WHEN 6 THEN @s_w_id6
                   WHEN 7 THEN @s_w_id7
                   WHEN 8 THEN @s_w_id8
                   WHEN 9 THEN @s_w_id9
                   WHEN 10 THEN @s_w_id10
                   WHEN 11 THEN @s_w_id11
                   WHEN 12 THEN @s_w_id12
                   WHEN 13 THEN @s_w_id13
                   WHEN 14 THEN @s_w_id14
                   WHEN 15 THEN @s_w_id15
                 END,

       @li_qty = CASE  @li_no
                   WHEN 1 THEN @ol_qty1
                   WHEN 2 THEN @ol_qty2
                   WHEN 3 THEN @ol_qty3
                   WHEN 4 THEN @ol_qty4
                   WHEN 5 THEN @ol_qty5
                   WHEN 6 THEN @ol_qty6
                   WHEN 7 THEN @ol_qty7
                   WHEN 8 THEN @ol_qty8
                   WHEN 9 THEN @ol_qty9
                   WHEN 10 THEN @ol_qty10
                   WHEN 11 THEN @ol_qty11
                   WHEN 12 THEN @ol_qty12
                   WHEN 13 THEN @ol_qty13
                   WHEN 14 THEN @ol_qty14
                   WHEN 15 THEN @ol_qty15
                 END
```

```
----------------------------------------
--  get item data (no one updates item)
----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
       UPDATE  stock
       SET     s_ytd           = s_ytd + @li_qty,
               @s_quantity     = s_quantity   = s_quantity - @li_qty +
                               CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
               s_order_cnt     = s_order_cnt + 1,
               s_remote_cnt    = s_remote_cnt +
                               CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
               @s_data         = s_data,
               @s_dist         = CASE @d_id
                               WHEN 1  THEN s_dist_01
                               WHEN 2  THEN s_dist_02
                               WHEN 3  THEN s_dist_03
                               WHEN 4  THEN s_dist_04
                               WHEN 5  THEN s_dist_05
                               WHEN 6  THEN s_dist_06
                               WHEN 7  THEN s_dist_07
                               WHEN 8  THEN s_dist_08
                               WHEN 9  THEN s_dist_09
                               WHEN 10 THEN s_dist_10
                              END
       WHERE   s_i_id          = @li_id AND
               s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
       IF (@@rowcount > 0)
       BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @s_dist)

----------------------------------
--  send line-item data to client
----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                       (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------

          SELECT  '',0,'',0,0
          SELECT  @commit_flag    = 0
                                  END
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
       SELECT  @c_last     = c_last,
               @c_discount = c_discount,
               @c_credit   = c_credit,
               @c_id_local = c_id
       FROM    customer WITH (repeatableread)
       WHERE   c_id        = @c_id AND
               c_w_id      = @w_id AND
               c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
       INSERT INTO orders VALUES ( @o_id,
                                   @d_id,
                                   @w_id,
                                   @c_id_local,
                                   0,
                                   @o_ol_cnt,
                                   @o_all_local,
                                   @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
       INSERT INTO new_order VALUES ( @o_id,
                                   @d_id,
                                   @w_id)

------------------------
--  select warehouse tax
------------------------
       SELECT  @w_tax  = w_tax
       FROM    warehouse WITH (repeatableread)
       WHERE   w_id    = @w_id

       IF (@commit_flag = 1)
                                   COMMIT TRANSACTION n
       ELSE
---------------------------------
--  all that work for nuthin!!!
---------------------------------
          ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
       SELECT  @w_tax,
               @d_tax,
               @o_id,
               @c_last,
               @c_discount,
               @c_credit,
               @o_entry_d,
               @commit_flag
END
GO
-- End SP8



-- Create SP9
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp9tpcc_neworder' )
   DROP PROCEDURE sp9tpcc_neworder
GO

CREATE PROC sp9tpcc_neworder
                   @w_id           int,
                   @d_id           tinyint,
                   @c_id           int,
                   @o_ol_cnt       tinyint,
                   @o_all_local    tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
```

```
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
-- get district tax and next availible order id and update
-- plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            @o_entry_d      = GETDATE(),
            d_next_o_id     = d_next_o_id + 1,
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
-- process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
        SELECT  @li_no  = @li_no + 1

------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
        SELECT  @li_id = CASE @li_no
                         WHEN 1 THEN @i_id1
                         WHEN 2 THEN @i_id2
                         WHEN 3 THEN @i_id3
                         WHEN 4 THEN @i_id4
                         WHEN 5 THEN @i_id5
                         WHEN 6 THEN @i_id6
                         WHEN 7 THEN @i_id7
                         WHEN 8 THEN @i_id8
                         WHEN 9 THEN @i_id9
                         WHEN 10 THEN @i_id10
                         WHEN 11 THEN @i_id11
                         WHEN 12 THEN @i_id12
                         WHEN 13 THEN @i_id13
                         WHEN 14 THEN @i_id14
                         WHEN 15 THEN @i_id15
                       END,

               @li_s_w_id = CASE @li_no
                         WHEN 1 THEN @s_w_id1
                         WHEN 2 THEN @s_w_id2
                         WHEN 3 THEN @s_w_id3
                         WHEN 4 THEN @s_w_id4
                         WHEN 5 THEN @s_w_id5
                         WHEN 6 THEN @s_w_id6
                         WHEN 7 THEN @s_w_id7
                         WHEN 8 THEN @s_w_id8
                         WHEN 9 THEN @s_w_id9
                         WHEN 10 THEN @s_w_id10
                         WHEN 11 THEN @s_w_id11
                         WHEN 12 THEN @s_w_id12
                         WHEN 13 THEN @s_w_id13
                         WHEN 14 THEN @s_w_id14
                         WHEN 15 THEN @s_w_id15
                       END,

               @li_qty = CASE  @li_no
                         WHEN 1 THEN @ol_qty1
                         WHEN 2 THEN @ol_qty2
                         WHEN 3 THEN @ol_qty3
                         WHEN 4 THEN @ol_qty4
                         WHEN 5 THEN @ol_qty5
                         WHEN 6 THEN @ol_qty6
                         WHEN 7 THEN @ol_qty7
                         WHEN 8 THEN @ol_qty8
                         WHEN 9 THEN @ol_qty9
                         WHEN 10 THEN @ol_qty10
                         WHEN 11 THEN @ol_qty11
                         WHEN 12 THEN @ol_qty12
                         WHEN 13 THEN @ol_qty13
                         WHEN 14 THEN @ol_qty14
                         WHEN 15 THEN @ol_qty15
                       END

----------------------------------------
-- get item data (no one updates item)
----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

------------------------
-- update stock values
------------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
                                  WHEN 3  THEN s_dist_03
                                  WHEN 4  THEN s_dist_04
                                  WHEN 5  THEN s_dist_05
                                  WHEN 6  THEN s_dist_06
                                  WHEN 7  THEN s_dist_07
                                  WHEN 8  THEN s_dist_08
                                  WHEN 9  THEN s_dist_09
                                  WHEN 10 THEN s_dist_10
                                  END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id
```

```
-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
       IF (@@rowcount > 0)
       BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

----------------------------------
--  send line-item data to client
----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
          SELECT  '',0,'',0,0
          SELECT  @commit_flag    = 0
                                END
               END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_d_id      = @d_id AND
           c_w_id      = @w_id AND

-----------------------------------------
--  insert fresh row into orders table
-----------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                               @d_id,
                               @w_id,
                               @c_id_local,
                               0,
                               @o_ol_cnt,
                               @o_all_local,
                               @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                                   @d_id,
                                   @w_id)

-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP9
```

```
IF EXISTS (SELECT name FROM sysobjects WHERE name =
'sp10tpcc_neworder' )
   DROP PROCEDURE sp10tpcc_neworder
GO

CREATE PROC sp10tpcc_neworder
                  @w_id           int,
                  @d_id           tinyint,
                  @c_id           int,
                  @o_ol_cnt       tinyint,
                  @o_all_local    tinyint,
                  @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                  @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                  @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                  @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                  @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                  @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                  @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                  @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                  @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                  @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                  @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                  @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                  @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                  @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                  @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
```

```
    @li_qty         smallint,
    @ol_number      int,
    @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

-----------------------
--  process orderlines
-----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
                       WHEN 6 THEN @ol_qty6
                       WHEN 7 THEN @ol_qty7
                       WHEN 8 THEN @ol_qty8
                       WHEN 9 THEN @ol_qty9
                       WHEN 10 THEN @ol_qty10
                       WHEN 11 THEN @ol_qty11
                       WHEN 12 THEN @ol_qty12
                       WHEN 13 THEN @ol_qty13
                       WHEN 14 THEN @ol_qty14
                       WHEN 15 THEN @ol_qty15
                     END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,          @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                      (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                              END
                              ELSE
                              BEGIN
-----------------------------------------------------------
```

```
-- no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                        END
            END
-------------------------------------------------------------
-- get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

---------------------------------------
-- insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-------------------------------------------------
-- insert corresponding row into new-order table
-------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                           @d_id,
                           @w_id)

-------------------------
-- select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
-- all that work for nuthin!!!
--------------------------------
      ROLLBACK TRANSACTION n

--------------------------------
-- return order data to client
--------------------------------
   SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP10


-- Create SP11
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp11tpcc_neworder' )
   DROP PROCEDURE sp11tpcc_neworder
GO

CREATE PROC sp11tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_last         char(16),
        @d_tax          smallmoney,
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
-- get district tax and next availible order id and update
-- plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
        @o_id           = d_next_o_id,
        d_next_o_id     = d_next_o_id + 1,
        @o_entry_d      = GETDATE(),
        @li_no          = 0,
        @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
        d_id            = @d_id

----------------------
-- process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
```

```
                WHEN 5 THEN @i_id5
                WHEN 6 THEN @i_id6
                WHEN 7 THEN @i_id7
                WHEN 8 THEN @i_id8
                WHEN 9 THEN @i_id9
                WHEN 10 THEN @i_id10
                WHEN 11 THEN @i_id11
                WHEN 12 THEN @i_id12
                WHEN 13 THEN @i_id13
                WHEN 14 THEN @i_id14
                WHEN 15 THEN @i_id15
            END,

        @li_s_w_id = CASE @li_no
                WHEN 1 THEN @s_w_id1
                WHEN 2 THEN @s_w_id2
                WHEN 3 THEN @s_w_id3
                WHEN 4 THEN @s_w_id4
                WHEN 5 THEN @s_w_id5
                WHEN 6 THEN @s_w_id6
                WHEN 7 THEN @s_w_id7
                WHEN 8 THEN @s_w_id8
                WHEN 9 THEN @s_w_id9
                WHEN 10 THEN @s_w_id10
                WHEN 11 THEN @s_w_id11
                WHEN 12 THEN @s_w_id12
                WHEN 13 THEN @s_w_id13
                WHEN 14 THEN @s_w_id14
                WHEN 15 THEN @s_w_id15
              END,

        @li_qty = CASE  @li_no
                WHEN 1 THEN @ol_qty1
                WHEN 2 THEN @ol_qty2
                WHEN 3 THEN @ol_qty3
                WHEN 4 THEN @ol_qty4
                WHEN 5 THEN @ol_qty5
                WHEN 6 THEN @ol_qty6
                WHEN 7 THEN @ol_qty7
                WHEN 8 THEN @ol_qty8
                WHEN 9 THEN @ol_qty9
                WHEN 10 THEN @ol_qty10
                WHEN 11 THEN @ol_qty11
                WHEN 12 THEN @ol_qty12
                WHEN 13 THEN @ol_qty13
                WHEN 14 THEN @ol_qty14
                WHEN 15 THEN @ol_qty15
              END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
        SELECT  @i_name  = i_name,
                @i_price = i_price,
                @i_data  = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id     = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd          = s_ytd + @li_qty,
                @s_quantity    = s_quantity   = s_quantity - @li_qty +
                                 CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt    = s_order_cnt + 1,
                s_remote_cnt   = s_remote_cnt +
                                 CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data        = s_data,
                @s_dist        = CASE @d_id
                                 WHEN 1  THEN s_dist_01
                                 WHEN 2  THEN s_dist_02
                                 WHEN 3  THEN s_dist_03
                                 WHEN 4  THEN s_dist_04
                                 WHEN 5  THEN s_dist_05
                                 WHEN 6  THEN s_dist_06
                                 WHEN 7  THEN s_dist_07
                                 WHEN 8  THEN s_dist_08
                                 WHEN 9  THEN s_dist_09
                                 WHEN 10 THEN s_dist_10
                                 END
        WHERE   s_i_id         = @li_id AND
                s_w_id         = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                         (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
            SELECT  ",0,",0,0
            SELECT  @commit_flag    = 0
                                END
                END
-------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)
```

```
-------------------------
-- select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
-- all that work for nuthin!!!
--------------------------------
      ROLLBACK TRANSACTION n

--------------------------------
-- return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP11



-- Create SP12
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp12tpcc_neworder' )
   DROP PROCEDURE sp12tpcc_neworder
GO

CREATE PROC sp12tpcc_neworder
                   @w_id           int,
                   @d_id           tinyint,
                   @c_id           int,
                   @o_ol_cnt       tinyint,
                   @o_all_local    tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                   @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                   @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                   @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                   @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                   @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
0,
                   @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                   @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                   @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                   @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                   @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

------------------------------------------------------------
-- get district tax and next availible order id and update
-- plus initialize local variables
------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id
-----------------------
-- process orderlines
-----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
```

```
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
    SELECT  @i_price    = i_price,
            @i_name     = i_name,
            @i_data     = i_data
    FROM    item WITH (repeatableread)
    WHERE   i_id        = @li_id

------------------------
--  update stock values
------------------------
    UPDATE  stock
    SET     s_ytd           = s_ytd + @li_qty,
            @s_quantity     = s_quantity   = s_quantity - @li_qty +
                        CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
            s_order_cnt     = s_order_cnt + 1,
            s_remote_cnt    = s_remote_cnt +
                        CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
            @s_data         = s_data,
            @s_dist         = CASE @d_id
                        WHEN 1  THEN s_dist_01
                        WHEN 2  THEN s_dist_02
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
    BEGIN
    IF (@@rowcount > 0)
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                     (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                                END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP12



-- Create SP13
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp13tpcc_neworder' )
    DROP PROCEDURE sp13tpcc_neworder
GO
```

```
CREATE PROC sp13tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0


AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

               @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
```

```
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
    IF (@@rowcount > 0)
    BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                        END
                        ELSE
                        BEGIN
-----------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-----------------------------------------------------------
        SELECT  '',0,'',0,0

        SELECT  @commit_flag    = 0
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

-----------------------------------------
--  insert fresh row into orders table
-----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @c_last,
            @c_id,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP13



-- Create SP14
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp14tpcc_neworder' )
    DROP PROCEDURE sp14tpcc_neworder
GO

CREATE PROC sp14tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
```

```
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

              @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

              @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

---------------------------------------
--  get item data (no one updates item)
---------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
       UPDATE  stock
       SET     s_ytd           = s_ytd + @li_qty,
               @s_quantity     = s_quantity   = s_quantity - @li_qty +
                               CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
               s_order_cnt     = s_order_cnt + 1,
               s_remote_cnt    = s_remote_cnt +
                               CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
               @s_data         = s_data,
               @s_dist         = CASE @d_id
                               WHEN 1  THEN s_dist_01
                               WHEN 2  THEN s_dist_02
                               WHEN 3  THEN s_dist_03
                               WHEN 4  THEN s_dist_04
                               WHEN 5  THEN s_dist_05
                               WHEN 6  THEN s_dist_06
                               WHEN 7  THEN s_dist_07
                               WHEN 8  THEN s_dist_08
                               WHEN 9  THEN s_dist_09
                               WHEN 10 THEN s_dist_10
                              END
       WHERE   s_i_id          = @li_id AND
               s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
       IF (@@rowcount > 0)
       BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
          SELECT  @i_name,
                  @s_quantity,
```

```
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
          SELECT  '',0,'',0,0
          SELECT  @commit_flag    = 0
                                END
               END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
          @c_discount = c_discount,
          @c_credit   = c_credit,
          @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
          c_w_id      = @w_id AND
          c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                           @d_id,
                           @w_id,
                           @c_id_local,
                           0,
                           @o_ol_cnt,
                           @o_all_local,
                           @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                              @d_id,
                              @w_id)

------------------------
--  select warehouse tax

   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                              COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
          @d_tax,
          @o_id,
          @c_last,
          @c_discount,
          @c_credit,
          @o_entry_d,
          @commit_flag
END
GO
-- End SP14



-- Create SP15
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp15tpcc_neworder' )
   DROP PROCEDURE sp15tpcc_neworder
GO

CREATE PROC sp15tpcc_neworder
                   @w_id           int,
                   @d_id           tinyint,
                   @c_id           int,
                   @o_ol_cnt       tinyint,
                   @o_all_local    tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                   @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                   @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                   @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                   @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                   @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                   @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,                 @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                   @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                   @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                   @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
       @d_tax          smallmoney,
       @c_last         char(16),
       @c_credit       char(2),
       @c_discount     smallmoney,
       @i_price        smallmoney,
       @i_name         char(24),
       @i_data         char(50),
       @o_entry_d      datetime,
       @remote_flag    int,
       @s_quantity     smallint,
       @s_data         char(50),
       @s_dist         char(24),
       @li_no          int,
       @o_id           int,
       @commit_flag    tinyint,
       @li_id          int,
       @li_s_w_id      int,
       @li_qty         smallint,
       @ol_number      int,
       @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
          @o_id           = d_next_o_id,
          d_next_o_id     = d_next_o_id + 1,
          @o_entry_d      = GETDATE(),
          @li_no          = 0,
          @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
          d_id            = @d_id

----------------------
```

```
-- process orderlines
-----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
      SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                      END,

             @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                      END,

             @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                      END

----------------------------------------
-- get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

----------------------
-- update stock values
----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-----------------------------------------------------------------
-- if there actually is a stock (and item) with these ids, go to work
-----------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------

---insert order_line data (using data from item and stock)
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

----------------------------------
-- send line-item data to client
----------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                 (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                                 END
                                 ELSE
                                 BEGIN
-------------------------------------------------------------
-- no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
         SELECT  ",0,",0,0
         SELECT  @commit_flag    = 0
                                 END
              END
-------------------------------------------------------
-- get customer last name, discount, and credit rating
-------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

----------------------------------
-- insert fresh row into orders table
----------------------------------
   INSERT INTO orders VALUES ( @o_id,
                               @d_id,
```

```
                    @w_id,
                    @c_id_local,
                    0,
                    @o_ol_cnt,
                    @o_all_local,
                    @o_entry_d)

------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                           @d_id,
                           @w_id)

------------------------
--  select warehouse tax
------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP15



-- Create SP16
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp16tpcc_neworder' )
   DROP PROCEDURE sp16tpcc_neworder
GO
```

```
CREATE PROC sp16tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN
```

```
BEGIN TRANSACTION n
------------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
------------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
```

```
                    WHEN 10 THEN @s_w_id10
                    WHEN 11 THEN @s_w_id11
                    WHEN 12 THEN @s_w_id12
                    WHEN 13 THEN @s_w_id13
                    WHEN 14 THEN @s_w_id14
                    WHEN 15 THEN @s_w_id15
                  END,

        @li_qty = CASE  @li_no
                    WHEN 1 THEN @ol_qty1
                    WHEN 2 THEN @ol_qty2
                    WHEN 3 THEN @ol_qty3
                    WHEN 4 THEN @ol_qty4
                    WHEN 5 THEN @ol_qty5
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                  END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                        CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                        CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                        WHEN 1  THEN s_dist_01
                        WHEN 2  THEN s_dist_02
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                      END
      WHERE   s_i_id        = @li_id AND
              s_w_id        = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
------------------------------------------------------------
--  insert order_line data (using data from item and stock)
------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                       (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

------------------------------------------------------------
          SELECT  ",0,",0,0
          SELECT  @commit_flag    = 0
                                  END
              END

--  get customer last name, discount, and credit rating
------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
```

```
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP16



-- Create SP17
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp17tpcc_neworder' )
   DROP PROCEDURE sp17tpcc_neworder
GO

CREATE PROC sp17tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @o_entry_d      datetime,
        @o_data         char(50),
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                END,

              @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                      END,

              @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
```

```
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                              CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                              CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                              WHEN 1  THEN s_dist_01
                              WHEN 2  THEN s_dist_02
                              WHEN 3  THEN s_dist_03
                              WHEN 4  THEN s_dist_04
                              WHEN 5  THEN s_dist_05
                              WHEN 6  THEN s_dist_06
                              WHEN 7  THEN s_dist_07
                              WHEN 8  THEN s_dist_08
                              WHEN 9  THEN s_dist_09
                              WHEN 10 THEN s_dist_10
                             END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                  @d_id,
                                  @w_id,
                                  @li_no,
                                  @li_id,
                                  'dec 31, 1899',
                                  @i_price * @li_qty,
                                  @li_s_w_id,
                                  @li_qty,
                                  @s_dist)

----------------------------------
--  send line-item data to client
----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                  (patindex('%ORIGINAL%',@s_data) > 0) )
THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
          SELECT  ",0,",0,0
          SELECT  @commit_flag    = 0
                                  END
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

-----------------------------------------
--  insert fresh row into orders table
-----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP17



-- Create SP18
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp18tpcc_neworder' )
    DROP PROCEDURE sp18tpcc_neworder
GO

CREATE PROC sp18tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
```

```
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
BEGIN
WHILE (@li_no < @o_ol_cnt)
    SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

                @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

                @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

----------------------
--  update stock values
----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                    @d_id,
```

```
                        @w_id,
                        @li_no,
                        @li_id,
                        'dec 31, 1899',
                        @i_price * @li_qty,
                        @li_s_w_id,
                        @li_qty,
                        @s_dist)

----------------------------------
--  send line-item data to client
----------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                        END
                        ELSE
                        BEGIN
------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

------------------------------------------------------------
        SELECT  ",0,",0,0
        SELECT  @commit_flag    = 0
                        END
                END
------------------------------------------------------------
--  get customer last name, discount, and credit rating
------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

------------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                                   @d_id,
                                   @w_id)

-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP18



-- Create SP19
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp19tpcc_neworder' )
   DROP PROCEDURE sp19tpcc_neworder
GO

CREATE PROC sp19tpcc_neworder
                   @w_id          int,
                   @d_id          tinyint,
                   @c_id          int,
                   @o_ol_cnt      tinyint,
                   @o_all_local   tinyint,
                   @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                   @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                   @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                   @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                   @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                   @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                   @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                   @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                   @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                   @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                   @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                   @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                   @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                   @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                   @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

------------------------------------------------------------
--  get district tax and next availible order id and update
```

```
-- plus initialize local variables
----------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
-- process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
                       WHEN 6 THEN @ol_qty6
                       WHEN 7 THEN @ol_qty7
                       WHEN 8 THEN @ol_qty8
                       WHEN 9 THEN @ol_qty9
                       WHEN 10 THEN @ol_qty10
                       WHEN 11 THEN @ol_qty11
                       WHEN 12 THEN @ol_qty12
                       WHEN 13 THEN @ol_qty13
                       WHEN 14 THEN @ol_qty14
                       WHEN 15 THEN @ol_qty15
                     END

----------------------------------------
-- get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

----------------------
-- update stock values
----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

----------------------------------------------------------------
-- if there actually is a stock (and item) with these ids, go to work
----------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
------------------------------------------------------------
-- insert order_line data (using data from item and stock)
------------------------------------------------------------
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

-----------------------------------
-- send line-item data to client
-----------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                      (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                                      END
                                      ELSE
                                      BEGIN
------------------------------------------------------------
-- no item (or stock) found - triggers rollback condition

------------------------------------------------------------
         SELECT  ",0,",0,0
         SELECT  @commit_flag    = 0
                                      END
            END
------------------------------------------------------------
-- get customer last name, discount, and credit rating
------------------------------------------------------------
   SELECT  @c_last     = c_last,
```

```
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

------------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                            @d_id,
                            @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                            COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,

END     @commit_flag
GO
-- End SP19



-- Create SP20
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp20tpcc_neworder' )
    DROP PROCEDURE sp20tpcc_neworder
GO

CREATE PROC sp20tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_dist         char(24),
        @s_data         char(50),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

-----------------------
--  process orderlines
-----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
        SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
        SELECT  @li_id = CASE @li_no
                            WHEN 1 THEN @i_id1
                            WHEN 2 THEN @i_id2
                            WHEN 3 THEN @i_id3
                            WHEN 4 THEN @i_id4
                            WHEN 5 THEN @i_id5
                            WHEN 6 THEN @i_id6
                            WHEN 7 THEN @i_id7
                            WHEN 8 THEN @i_id8
                            WHEN 9 THEN @i_id9
                            WHEN 10 THEN @i_id10
                            WHEN 11 THEN @i_id11
                            WHEN 12 THEN @i_id12
                            WHEN 13 THEN @i_id13
                            WHEN 14 THEN @i_id14
```

```
                WHEN 15 THEN @i_id15
            END,

        @li_s_w_id = CASE @li_no
                WHEN 1 THEN @s_w_id1
                WHEN 2 THEN @s_w_id2
                WHEN 3 THEN @s_w_id3
                WHEN 4 THEN @s_w_id4
                WHEN 5 THEN @s_w_id5
                WHEN 6 THEN @s_w_id6
                WHEN 7 THEN @s_w_id7
                WHEN 8 THEN @s_w_id8
                WHEN 9 THEN @s_w_id9
                WHEN 10 THEN @s_w_id10
                WHEN 11 THEN @s_w_id11
                WHEN 12 THEN @s_w_id12
                WHEN 13 THEN @s_w_id13
                WHEN 14 THEN @s_w_id14
                WHEN 15 THEN @s_w_id15
            END,

        @li_qty = CASE  @li_no
                WHEN 1 THEN @ol_qty1
                WHEN 2 THEN @ol_qty2
                WHEN 3 THEN @ol_qty3
                WHEN 4 THEN @ol_qty4
                WHEN 5 THEN @ol_qty5
                WHEN 6 THEN @ol_qty6
                WHEN 7 THEN @ol_qty7
                WHEN 8 THEN @ol_qty8
                WHEN 9 THEN @ol_qty9
                WHEN 10 THEN @ol_qty10
                WHEN 11 THEN @ol_qty11
                WHEN 12 THEN @ol_qty12
                WHEN 13 THEN @ol_qty13
                WHEN 14 THEN @ol_qty14
                WHEN 15 THEN @ol_qty15
            END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
    SELECT  @i_price    = i_price,
            @i_name     = i_name,
            @i_data     = i_data
    FROM    item WITH (repeatableread)
    WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
    UPDATE  stock
    SET     @s_quantity   = s_quantity,
            s_ytd         = s_ytd + @li_qty,
            s_quantity    = s_quantity - @li_qty +
                            CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
            s_order_cnt   = s_order_cnt + 1,
            s_remote_cnt  = s_remote_cnt +
                            CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
            @s_data       = s_data,
            @s_dist       = CASE @d_id
                            WHEN 1  THEN s_dist_01
                            WHEN 2  THEN s_dist_02
                            WHEN 3  THEN s_dist_03
                            WHEN 4  THEN s_dist_04
                            WHEN 5  THEN s_dist_05
                            WHEN 6  THEN s_dist_06
                            WHEN 7  THEN s_dist_07
                            WHEN 8  THEN s_dist_08
                            WHEN 9  THEN s_dist_09
                            WHEN 10 THEN s_dist_10
                            END
    WHERE   s_i_id        = @li_id AND
            s_w_id        = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
    IF (@@rowcount > 0)
    BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                     (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price * @li_qty
                        END
                        ELSE
                        BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  ",0,",0,0
        SELECT  @commit_flag    = 0
                        END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                            @d_id,
                            @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                            COMMIT TRANSACTION n
    ELSE
```

```
---------------------------------
--  all that work for nuthin!!!
---------------------------------
        ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP20

-- Create SP21
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp21tpcc_neworder' )
    DROP PROCEDURE sp21tpcc_neworder
GO

CREATE PROC sp21tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
SELECT  @li_id = CASE @li_no
                    WHEN 1 THEN @i_id1
                    WHEN 2 THEN @i_id2
                    WHEN 3 THEN @i_id3
                    WHEN 4 THEN @i_id4
                    WHEN 5 THEN @i_id5
                    WHEN 6 THEN @i_id6
                    WHEN 7 THEN @i_id7
                    WHEN 8 THEN @i_id8
                    WHEN 9 THEN @i_id9
                    WHEN 10 THEN @i_id10
                    WHEN 11 THEN @i_id11
                    WHEN 12 THEN @i_id12
                    WHEN 13 THEN @i_id13
                    WHEN 14 THEN @i_id14
                    WHEN 15 THEN @i_id15
                END,

       @li_s_w_id = CASE @li_no
                    WHEN 1 THEN @s_w_id1
                    WHEN 2 THEN @s_w_id2
                    WHEN 3 THEN @s_w_id3
                    WHEN 4 THEN @s_w_id4
                    WHEN 5 THEN @s_w_id5
                    WHEN 6 THEN @s_w_id6
                    WHEN 7 THEN @s_w_id7
                    WHEN 8 THEN @s_w_id8
                    WHEN 9 THEN @s_w_id9
                    WHEN 10 THEN @s_w_id10
                    WHEN 11 THEN @s_w_id11
                    WHEN 12 THEN @s_w_id12
                    WHEN 13 THEN @s_w_id13
                    WHEN 14 THEN @s_w_id14
                    WHEN 15 THEN @s_w_id15
                END,

       @li_qty = CASE  @li_no
                    WHEN 1 THEN @ol_qty1
                    WHEN 2 THEN @ol_qty2
                    WHEN 3 THEN @ol_qty3
                    WHEN 4 THEN @ol_qty4
                    WHEN 5 THEN @ol_qty5
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                END
```

```
-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                              CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                              CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                              WHEN 1  THEN s_dist_01
                              WHEN 2  THEN s_dist_02
                              WHEN 3  THEN s_dist_03
                              WHEN 4  THEN s_dist_04
                              WHEN 5  THEN s_dist_05
                              WHEN 6  THEN s_dist_06
                              WHEN 7  THEN s_dist_07
                              WHEN 8  THEN s_dist_08
                              WHEN 9  THEN s_dist_09
                              WHEN 10 THEN s_dist_10
                             END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
              SELECT  @i_name,
                      @s_quantity,
                      b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                   (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                      @i_price,
                      @i_price * @li_qty
                                   END
                                   ELSE
                                   BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-------------------------------------------------------------
              SELECT  '',0,'',0,0
              SELECT  @commit_flag    = 0
                                   END
                  END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
      SELECT  @c_last     = c_last,
              @c_discount = c_discount,
              @c_credit   = c_credit,
              @c_id_local = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
      INSERT INTO orders VALUES ( @o_id,
                                  @d_id,
                                  @w_id,
                                  @c_id_local,
                                  0,
                                  @o_ol_cnt,
                                  @o_all_local,
                                  @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
      INSERT INTO new_order VALUES ( @o_id,
                                  @d_id,
                                  @w_id)

------------------------
--  select warehouse tax
------------------------
      SELECT  @w_tax  = w_tax
      FROM    warehouse WITH (repeatableread)
      WHERE   w_id    = @w_id

      IF (@commit_flag = 1)
                                  COMMIT TRANSACTION n
      ELSE
---------------------------------
--  all that work for nuthin!!!
---------------------------------
          ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
      SELECT  @w_tax,
              @d_tax,
              @o_id,
              @c_last,
              @c_discount,
              @c_credit,
              @o_entry_d,
              @commit_flag
END
GO
-- End SP21



-- Create SP22
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp22tpcc_neworder' )
    DROP PROCEDURE sp22tpcc_neworder
GO

CREATE PROC sp22tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
```

```
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0


AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
-- get district tax and next availible order id and update
-- plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            @o_entry_d      = GETDATE(),
            d_next_o_id     = d_next_o_id + 1,
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
-- process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
        SELECT  @li_no  = @li_no + 1

------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
        SELECT  @li_id = CASE @li_no
                         WHEN 1 THEN @i_id1
                         WHEN 2 THEN @i_id2
                         WHEN 3 THEN @i_id3
                         WHEN 4 THEN @i_id4
                         WHEN 5 THEN @i_id5
                         WHEN 6 THEN @i_id6
                         WHEN 7 THEN @i_id7
                         WHEN 8 THEN @i_id8
                         WHEN 9 THEN @i_id9
                         WHEN 10 THEN @i_id10
                         WHEN 11 THEN @i_id11
                         WHEN 12 THEN @i_id12
                         WHEN 13 THEN @i_id13
                         WHEN 14 THEN @i_id14
                         WHEN 15 THEN @i_id15
                       END,

               @li_s_w_id = CASE @li_no
                         WHEN 1 THEN @s_w_id1
                         WHEN 2 THEN @s_w_id2
                         WHEN 3 THEN @s_w_id3
                         WHEN 4 THEN @s_w_id4
                         WHEN 5 THEN @s_w_id5
                         WHEN 6 THEN @s_w_id6
                         WHEN 7 THEN @s_w_id7
                         WHEN 8 THEN @s_w_id8
                         WHEN 9 THEN @s_w_id9
                         WHEN 10 THEN @s_w_id10
                         WHEN 11 THEN @s_w_id11
                         WHEN 12 THEN @s_w_id12
                         WHEN 13 THEN @s_w_id13
                         WHEN 14 THEN @s_w_id14
                         WHEN 15 THEN @s_w_id15
                       END,

               @li_qty = CASE  @li_no
                         WHEN 1 THEN @ol_qty1
                         WHEN 2 THEN @ol_qty2
                         WHEN 3 THEN @ol_qty3
                         WHEN 4 THEN @ol_qty4
                         WHEN 5 THEN @ol_qty5
                         WHEN 6 THEN @ol_qty6
                         WHEN 7 THEN @ol_qty7
                         WHEN 8 THEN @ol_qty8
                         WHEN 9 THEN @ol_qty9
                         WHEN 10 THEN @ol_qty10
                         WHEN 11 THEN @ol_qty11
                         WHEN 12 THEN @ol_qty12
                         WHEN 13 THEN @ol_qty13
                         WHEN 14 THEN @ol_qty14
                         WHEN 15 THEN @ol_qty15
                       END

----------------------------------------
-- get item data (no one updates item)
----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

------------------------
-- update stock values
------------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
                                  WHEN 3  THEN s_dist_03
                                  WHEN 4  THEN s_dist_04
                                  WHEN 5  THEN s_dist_05
                                  WHEN 6  THEN s_dist_06
                                  WHEN 7  THEN s_dist_07
                                  WHEN 8  THEN s_dist_08
                                  WHEN 9  THEN s_dist_09
                                  WHEN 10 THEN s_dist_10
                                  END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id
```

```
-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
       IF (@@rowcount > 0)
       BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
          SELECT  '',0,'',0,0
          SELECT  @commit_flag    = 0
                                  END
               END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_d_id      = @d_id
           c_w_id      = @w_id AND

-----------------------------------------
--  insert fresh row into orders table
-----------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                               @d_id,
                               @w_id,
                               @c_id_local,
                               0,
                               @o_ol_cnt,
                               @o_all_local,
                               @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                                   @d_id,
                                   @w_id)

-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP22
```

```
IF EXISTS (SELECT name FROM sysobjects WHERE name =
'sp23tpcc_neworder' )
   DROP PROCEDURE sp23tpcc_neworder
GO

CREATE PROC sp23tpcc_neworder
                  @w_id           int,
                  @d_id           tinyint,
                  @c_id           int,
                  @o_ol_cnt       tinyint,
                  @o_all_local    tinyint,
                  @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                  @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                  @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                  @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                  @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                  @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                  @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                  @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                  @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                  @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                  @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                  @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                  @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                  @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                  @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
```

```
    @li_qty         smallint,
    @ol_number      int,
    @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
                       WHEN 5 THEN @ol_qty5
                       WHEN 6 THEN @ol_qty6
                       WHEN 7 THEN @ol_qty7
                       WHEN 8 THEN @ol_qty8
                       WHEN 9 THEN @ol_qty9
                       WHEN 10 THEN @ol_qty10
                       WHEN 11 THEN @ol_qty11
                       WHEN 12 THEN @ol_qty12
                       WHEN 13 THEN @ol_qty13
                       WHEN 14 THEN @ol_qty14
                       WHEN 15 THEN @ol_qty15
                     END

---------------------------------------
--  get item data (no one updates item)
---------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

----------------------
--  update stock values
----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,          @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                      (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                          END
                          ELSE
                          BEGIN
-------------------------------------------------------------
```

```
-- no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                        END
            END
-------------------------------------------------------------
-- get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

---------------------------------------
-- insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-------------------------------------------------
-- insert corresponding row into new-order table
-------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                           @d_id,
                           @w_id)

------------------------
-- select warehouse tax
------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
-------------------------------
-- all that work for nuthin!!!
-------------------------------
      ROLLBACK TRANSACTION n

--------------------------------
-- return order data to client
--------------------------------
   SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP23



-- Create SP24
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp24tpcc_neworder' )
   DROP PROCEDURE sp24tpcc_neworder
GO

CREATE PROC sp24tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @d_tax          smallmoney,
        @w_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
-- get district tax and next availible order id and update
-- plus initialize local variables
-----------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
        @o_id           = d_next_o_id,
        d_next_o_id     = d_next_o_id + 1,
        @o_entry_d      = GETDATE(),
        @li_no          = 0,
        @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
        d_id            = @d_id

----------------------
-- process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
```

```
                WHEN 4 THEN @i_id4
                WHEN 5 THEN @i_id5
                WHEN 6 THEN @i_id6
                WHEN 7 THEN @i_id7
                WHEN 8 THEN @i_id8
                WHEN 9 THEN @i_id9
                WHEN 10 THEN @i_id10
                WHEN 11 THEN @i_id11
                WHEN 12 THEN @i_id12
                WHEN 13 THEN @i_id13
                WHEN 14 THEN @i_id14
                WHEN 15 THEN @i_id15
            END,

    @li_s_w_id = CASE @li_no
                WHEN 1 THEN @s_w_id1
                WHEN 2 THEN @s_w_id2
                WHEN 3 THEN @s_w_id3
                WHEN 4 THEN @s_w_id4
                WHEN 5 THEN @s_w_id5
                WHEN 6 THEN @s_w_id6
                WHEN 7 THEN @s_w_id7
                WHEN 8 THEN @s_w_id8
                WHEN 9 THEN @s_w_id9
                WHEN 10 THEN @s_w_id10
                WHEN 11 THEN @s_w_id11
                WHEN 12 THEN @s_w_id12
                WHEN 13 THEN @s_w_id13
                WHEN 14 THEN @s_w_id14
                WHEN 15 THEN @s_w_id15
               END,

    @li_qty = CASE  @li_no
                WHEN 1 THEN @ol_qty1
                WHEN 2 THEN @ol_qty2
                WHEN 3 THEN @ol_qty3
                WHEN 4 THEN @ol_qty4
                WHEN 5 THEN @ol_qty5
                WHEN 6 THEN @ol_qty6
                WHEN 7 THEN @ol_qty7
                WHEN 8 THEN @ol_qty8
                WHEN 9 THEN @ol_qty9
                WHEN 10 THEN @ol_qty10
                WHEN 11 THEN @ol_qty11
                WHEN 12 THEN @ol_qty12
                WHEN 13 THEN @ol_qty13
                WHEN 14 THEN @ol_qty14
                WHEN 15 THEN @ol_qty15
              END

----------------------------------------
--  get item data (no one updates item)
------ SELECT  @i_price  = i_price,
            @i_name     = i_name,
            @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                  CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                  CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                  WHEN 1  THEN s_dist_01
                                  WHEN 2  THEN s_dist_02
                                  WHEN 3  THEN s_dist_03
                                  WHEN 4  THEN s_dist_04
                                  WHEN 5  THEN s_dist_05
                                  WHEN 6  THEN s_dist_06
                                  WHEN 7  THEN s_dist_07
                                  WHEN 8  THEN s_dist_08
                                  WHEN 9  THEN s_dist_09
                                  WHEN 10 THEN s_dist_10
                                 END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)
-- send line-item data to client
-----------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                         (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
            SELECT  ",0,",0,0
            SELECT  @commit_flag    = 0
                                END
                END
-------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)
```

```
-------------------------
--  select warehouse tax
-------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP24



-- Create SP25
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp25tpcc_neworder' )
   DROP PROCEDURE sp25tpcc_neworder
GO

CREATE PROC sp25tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_id  = @d_id AND d_w_id  = @w_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
      SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                     END,

             @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
                       WHEN 10 THEN @s_w_id10
                       WHEN 11 THEN @s_w_id11
                       WHEN 12 THEN @s_w_id12
                       WHEN 13 THEN @s_w_id13
                       WHEN 14 THEN @s_w_id14
                       WHEN 15 THEN @s_w_id15
                     END,

             @li_qty = CASE  @li_no
                       WHEN 1 THEN @ol_qty1
                       WHEN 2 THEN @ol_qty2
                       WHEN 3 THEN @ol_qty3
                       WHEN 4 THEN @ol_qty4
```

```
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
    SELECT  @i_price    = i_price,
            @i_name     = i_name,
            @i_data     = i_data
    FROM    item WITH (repeatableread)
    WHERE   i_id        = @li_id

------------------------
--  update stock values
------------------------
    UPDATE  stock
    SET     s_ytd           = s_ytd + @li_qty,
            @s_quantity     = s_quantity   = s_quantity - @li_qty +
                        CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
            s_order_cnt     = s_order_cnt + 1,
            s_remote_cnt    = s_remote_cnt +
                        CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
            @s_data         = s_data,
            @s_dist         = CASE @d_id
                        WHEN 1  THEN s_dist_01
                        WHEN 2  THEN s_dist_02
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work

--------IF (@@rowcount > 0)-------------------------------------
        BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)


---------------------------------
--  send line-item data to client
---------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
            SELECT  '',0,'',0,0
            SELECT  @commit_flag    = 0
                                END
                END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id


----------------------------------------

--- insert fresh row into orders table
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
---------------------------------
--  all that work for nuthin!!!
---------------------------------
        ROLLBACK TRANSACTION n

---------------------------------
--  return order data to client
---------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP25



-- Create SP26
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp26tpcc_neworder' )
    DROP PROCEDURE sp26tpcc_neworder
```

```
GO

CREATE PROC sp26tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int
BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

              @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

              @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
       UPDATE  stock
       SET     s_ytd           = s_ytd + @li_qty,
               @s_quantity     = s_quantity   = s_quantity - @li_qty +
                               CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
               s_order_cnt     = s_order_cnt + 1,
               s_remote_cnt    = s_remote_cnt +
                               CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
               @s_data         = s_data,
               @s_dist         = CASE @d_id
                               WHEN 1  THEN s_dist_01
```

```
                        WHEN 2  THEN s_dist_02
                        WHEN 3  THEN s_dist_03
                        WHEN 4  THEN s_dist_04
                        WHEN 5  THEN s_dist_05
                        WHEN 6  THEN s_dist_06
                        WHEN 7  THEN s_dist_07
                        WHEN 8  THEN s_dist_08
                        WHEN 9  THEN s_dist_09
                        WHEN 10 THEN s_dist_10
                       END
    WHERE   s_i_id          = @li_id AND
            s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
    IF (@@rowcount > 0)
    BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
        INSERT INTO order_line VALUES(  @o_id,
                                        @d_id,
                                        @w_id,
                                        @li_no,
                                        @li_id,
                                        'dec 31, 1899',
                                        @i_price * @li_qty,
                                        @li_s_w_id,
                                        @li_qty,
                                        @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
                SELECT  @commit_flag    = 0
                                END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO
-- End SP26



-- Create SP27
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp27tpcc_neworder' )
    DROP PROCEDURE sp27tpcc_neworder
GO

CREATE PROC sp27tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
```

```
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

               @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id
-----------------------
--  update stock values
-----------------------
       UPDATE  stock
       SET     s_ytd           = s_ytd + @li_qty,
               @s_quantity     = s_quantity   = s_quantity - @li_qty +
                               CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
               s_order_cnt     = s_order_cnt + 1,
               s_remote_cnt    = s_remote_cnt +
                               CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
               @s_data         = s_data,
               @s_dist         = CASE @d_id
                               WHEN 1  THEN s_dist_01
                               WHEN 2  THEN s_dist_02
                               WHEN 3  THEN s_dist_03
                               WHEN 4  THEN s_dist_04
                               WHEN 5  THEN s_dist_05
                               WHEN 6  THEN s_dist_06
                               WHEN 7  THEN s_dist_07
                               WHEN 8  THEN s_dist_08
                               WHEN 9  THEN s_dist_09
                               WHEN 10 THEN s_dist_10
                               END
       WHERE   s_i_id          = @li_id AND
               s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
       IF (@@rowcount > 0)
       BEGIN
-----------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------
           INSERT INTO order_line VALUES(  @o_id,
                                   @d_id,
                                   @w_id,
                                   @li_no,
                                   @li_id,
                                   'dec 31, 1899',
                                   @i_price * @li_qty,
                                   @li_s_w_id,
                                   @li_qty,
                                   @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
           SELECT  @i_name,
                   @s_quantity,
```

```
            b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                        (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
            @i_price,
            @i_price * @li_qty
                        END
                        ELSE
                        BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                        END
            END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
        @c_discount = c_discount,
        @c_credit   = c_credit,
        @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
        c_w_id      = @w_id AND
        c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                        @d_id,
                        @w_id,
                        @c_id_local,
                        0,
                        @o_ol_cnt,
                        @o_all_local,
                        @o_entry_d)

-----------------------------------------------------
--  insert corresponding row into new-order table
-----------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                        @d_id,
                        @w_id)

------------------------
--  select warehouse tax

   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP27



-- Create SP28
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp28tpcc_neworder' )
   DROP PROCEDURE sp28tpcc_neworder
GO

CREATE PROC sp28tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
        @o_id           = d_next_o_id,
        d_next_o_id     = d_next_o_id + 1,
        @o_entry_d      = GETDATE(),
        @li_no          = 0,
        @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
        d_id            = @d_id

----------------------
```

```
-- process orderlines
-----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

------------------------------------------------
-- set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
      SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                      END,

             @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                      END,

             @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                      END

---------------------------------------
-- get item data (no one updates item)
---------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

----------------------
-- update stock values
----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-----------------------------------------------------------------------
-- if there actually is a stock (and item) with these ids, go to work
-----------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------

---insert order_line data (using data from item and stock)
         INSERT INTO order_line VALUES(  @o_id,
                                         @d_id,
                                         @w_id,
                                         @li_no,
                                         @li_id,
                                         'dec 31, 1899',
                                         @i_price * @li_qty,
                                         @li_s_w_id,
                                         @li_qty,
                                         @s_dist)

---------------------------------
-- send line-item data to client
---------------------------------
         SELECT  @i_name,
                 @s_quantity,
                 b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                 (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                 @i_price,
                 @i_price * @li_qty
                                 END
                                 ELSE
                                 BEGIN
-------------------------------------------------------------
-- no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
         SELECT  ",0,",0,0
         SELECT  @commit_flag    = 0
                                 END
              END
-------------------------------------------------------
-- get customer last name, discount, and credit rating
-------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

---------------------------------
-- insert fresh row into orders table
---------------------------------
   INSERT INTO orders VALUES ( @o_id,
                               @d_id,
```

```
                    @w_id,
                    @c_id_local,
                    0,
                    @o_ol_cnt,
                    @o_all_local,
                    @o_entry_d)

------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                        @d_id,
                        @w_id)

-----------------------
--  select warehouse tax
-----------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                        COMMIT TRANSACTION n
   ELSE
-------------------------------
--  all that work for nuthin!!!
-------------------------------
       ROLLBACK TRANSACTION n

-------------------------------
--  return order data to client
-------------------------------
   SELECT  @w_tax,
          @d_tax,
          @o_id,
          @c_last,
          @c_discount,
          @c_credit,
          @o_entry_d,
          @commit_flag
END
GO
-- End SP28



-- Create SP29
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp29tpcc_neworder' )
   DROP PROCEDURE sp29tpcc_neworder
GO
```

```
CREATE PROC sp29tpcc_neworder
                    @w_id           int,
                    @d_id           tinyint,
                    @c_id           int,
                    @o_ol_cnt       tinyint,
                    @o_all_local    tinyint,
                    @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                    @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                    @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                    @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                    @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                    @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                    @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                    @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                    @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
       @d_tax          smallmoney,
       @c_last         char(16),
       @c_credit       char(2),
       @c_discount     smallmoney,
       @i_price        smallmoney,
       @i_name         char(24),
       @i_data         char(50),
       @o_entry_d      datetime,
       @remote_flag    int,
       @s_quantity     smallint,
       @s_data         char(50),
       @s_dist         char(24),
       @li_no          int,
       @o_id           int,
       @commit_flag    tinyint,
       @li_id          int,
       @li_s_w_id      int,
       @li_qty         smallint,
       @ol_number      int,
       @c_id_local     int

BEGIN
```

```
BEGIN TRANSACTION n
---------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
---------------------------------------------------------------
   UPDATE  district
   SET     @d_tax          = d_tax,
           @o_id           = d_next_o_id,
           d_next_o_id     = d_next_o_id + 1,
           @o_entry_d      = GETDATE(),
           @li_no          = 0,
           @commit_flag    = 1
   WHERE   d_w_id          = @w_id AND
           d_id            = @d_id

----------------------
--  process orderlines
----------------------
   WHILE (@li_no < @o_ol_cnt)
   BEGIN
      SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
       SELECT  @li_id = CASE @li_no
                       WHEN 1 THEN @i_id1
                       WHEN 2 THEN @i_id2
                       WHEN 3 THEN @i_id3
                       WHEN 4 THEN @i_id4
                       WHEN 5 THEN @i_id5
                       WHEN 6 THEN @i_id6
                       WHEN 7 THEN @i_id7
                       WHEN 8 THEN @i_id8
                       WHEN 9 THEN @i_id9
                       WHEN 10 THEN @i_id10
                       WHEN 11 THEN @i_id11
                       WHEN 12 THEN @i_id12
                       WHEN 13 THEN @i_id13
                       WHEN 14 THEN @i_id14
                       WHEN 15 THEN @i_id15
                   END,

              @li_s_w_id = CASE @li_no
                       WHEN 1 THEN @s_w_id1
                       WHEN 2 THEN @s_w_id2
                       WHEN 3 THEN @s_w_id3
                       WHEN 4 THEN @s_w_id4
                       WHEN 5 THEN @s_w_id5
                       WHEN 6 THEN @s_w_id6
                       WHEN 7 THEN @s_w_id7
                       WHEN 8 THEN @s_w_id8
                       WHEN 9 THEN @s_w_id9
```

```
                    WHEN 10 THEN @s_w_id10
                    WHEN 11 THEN @s_w_id11
                    WHEN 12 THEN @s_w_id12
                    WHEN 13 THEN @s_w_id13
                    WHEN 14 THEN @s_w_id14
                    WHEN 15 THEN @s_w_id15
                  END,

        @li_qty = CASE  @li_no
                    WHEN 1 THEN @ol_qty1
                    WHEN 2 THEN @ol_qty2
                    WHEN 3 THEN @ol_qty3
                    WHEN 4 THEN @ol_qty4
                    WHEN 5 THEN @ol_qty5
                    WHEN 6 THEN @ol_qty6
                    WHEN 7 THEN @ol_qty7
                    WHEN 8 THEN @ol_qty8
                    WHEN 9 THEN @ol_qty9
                    WHEN 10 THEN @ol_qty10
                    WHEN 11 THEN @ol_qty11
                    WHEN 12 THEN @ol_qty12
                    WHEN 13 THEN @ol_qty13
                    WHEN 14 THEN @ol_qty14
                    WHEN 15 THEN @ol_qty15
                  END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
      SELECT  @i_price    = i_price,
              @i_name     = i_name,
              @i_data     = i_data
      FROM    item WITH (repeatableread)
      WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
      UPDATE  stock
      SET     s_ytd           = s_ytd + @li_qty,
              @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt     = s_order_cnt + 1,
              s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data         = s_data,
              @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                              END
      WHERE   s_i_id          = @li_id AND
              s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
------------------------------------------------------------
--  insert order_line data (using data from item and stock)
------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

-----------------------------------
--  send line-item data to client
-----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                       (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                  END
                                  ELSE
                                  BEGIN
------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

------------------------------------------------------------
          SELECT  '',0,'',0,0
          SELECT  @commit_flag    = 0
                                  END
              END

--  get customer last name, discount, and credit rating
------------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

------------------------------------------------
--  insert corresponding row into new-order table
------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                                COMMIT TRANSACTION n
    ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
        ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
```

```
        @c_credit,
        @o_entry_d,
        @commit_flag
END
GO
-- End SP29



-- Create SP30
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp30tpcc_neworder' )
   DROP PROCEDURE sp30tpcc_neworder
GO

CREATE PROC sp30tpcc_neworder
                @w_id           int,
                @d_id           tinyint,
                @c_id           int,
                @o_ol_cnt       tinyint,
                @o_all_local    tinyint,
                @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @o_entry_d      datetime,
        @s_data         char(50),
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
       SELECT  @li_no  = @li_no + 1

------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
------------------------------------------------
       SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                END,

               @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                      END,

               @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                END

----------------------------------------
--  get item data (no one updates item)
----------------------------------------
       SELECT  @i_price    = i_price,
               @i_name     = i_name,
               @i_data     = i_data
       FROM    item WITH (repeatableread)
       WHERE   i_id        = @li_id

-----------------------
--  update stock values
```

```
-----------------------
      UPDATE  stock
      SET     s_ytd          = s_ytd + @li_qty,
              @s_quantity    = s_quantity   = s_quantity - @li_qty +
                               CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
              s_order_cnt    = s_order_cnt + 1,
              s_remote_cnt   = s_remote_cnt +
                               CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
              @s_data        = s_data,
              @s_dist        = CASE @d_id
                               WHEN 1  THEN s_dist_01
                               WHEN 2  THEN s_dist_02
                               WHEN 3  THEN s_dist_03
                               WHEN 4  THEN s_dist_04
                               WHEN 5  THEN s_dist_05
                               WHEN 6  THEN s_dist_06
                               WHEN 7  THEN s_dist_07
                               WHEN 8  THEN s_dist_08
                               WHEN 9  THEN s_dist_09
                               WHEN 10 THEN s_dist_10
                              END
      WHERE   s_i_id         = @li_id AND
              s_w_id         = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
      IF (@@rowcount > 0)
      BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
          INSERT INTO order_line VALUES(  @o_id,
                                          @d_id,
                                          @w_id,
                                          @li_no,
                                          @li_id,
                                          'dec 31, 1899',
                                          @i_price * @li_qty,
                                          @li_s_w_id,
                                          @li_qty,
                                          @s_dist)

----------------------------------
--  send line-item data to client
----------------------------------
          SELECT  @i_name,
                  @s_quantity,
                  b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND
                                       (patindex('%ORIGINAL%',@s_data) > 0) )
THEN 'B' ELSE 'G' END,
                  @i_price,
                  @i_price * @li_qty
                                       END
                                       ELSE
                                       BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
          SELECT  ",0,",0,0
          SELECT  @commit_flag    = 0
                                       END
                  END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
      SELECT  @c_last     = c_last,
              @c_discount = c_discount,
              @c_credit   = c_credit,
              @c_id_local = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id


----------------------------------------
--  insert fresh row into orders table
----------------------------------------
      INSERT INTO orders VALUES ( @o_id,
                                  @d_id,
                                  @w_id,
                                  @c_id_local,
                                  0,
                                  @o_ol_cnt,
                                  @o_all_local,
                                  @o_entry_d)


-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
      INSERT INTO new_order VALUES (  @o_id,
                                      @d_id,
                                      @w_id)


------------------------
--  select warehouse tax
------------------------
      SELECT  @w_tax  = w_tax
      FROM    warehouse WITH (repeatableread)
      WHERE   w_id    = @w_id

      IF (@commit_flag = 1)
                                  COMMIT TRANSACTION n
      ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
          ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
      SELECT  @w_tax,
              @d_tax,
              @o_id,
              @c_last,
              @c_discount,
              @c_credit,
              @o_entry_d,
              @commit_flag
END
GO
-- End SP30


-- Create SP31
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp31tpcc_neworder' )
    DROP PROCEDURE sp31tpcc_neworder
GO

CREATE PROC sp31tpcc_neworder
                  @w_id           int,
                  @d_id           tinyint,
                  @c_id           int,
                  @o_ol_cnt       tinyint,
                  @o_all_local    tinyint,
                  @i_id1  int = 0, @s_w_id1  int = 0, @ol_qty1  smallint = 0,
                  @i_id2  int = 0, @s_w_id2  int = 0, @ol_qty2  smallint = 0,
                  @i_id3  int = 0, @s_w_id3  int = 0, @ol_qty3  smallint = 0,
                  @i_id4  int = 0, @s_w_id4  int = 0, @ol_qty4  smallint = 0,
                  @i_id5  int = 0, @s_w_id5  int = 0, @ol_qty5  smallint = 0,
                  @i_id6  int = 0, @s_w_id6  int = 0, @ol_qty6  smallint = 0,
                  @i_id7  int = 0, @s_w_id7  int = 0, @ol_qty7  smallint = 0,
                  @i_id8  int = 0, @s_w_id8  int = 0, @ol_qty8  smallint = 0,
                  @i_id9  int = 0, @s_w_id9  int = 0, @ol_qty9  smallint = 0,
                  @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint =
0,
                  @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint =
0,
                  @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint =
0,
                  @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint =
0,
```

```
                @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint =
0,
                @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0



AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     smallmoney,
        @i_price        smallmoney,
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      int,
        @li_qty         smallint,
        @ol_number      int,
        @c_id_local     int

BEGIN

BEGIN TRANSACTION n

-----------------------------------------------------------------
--  get district tax and next availible order id and update
--  plus initialize local variables
-----------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN

        SELECT  @li_no  = @li_no + 1
-------------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-------------------------------------------------
        SELECT  @li_id = CASE @li_no
                        WHEN 1 THEN @i_id1
                        WHEN 2 THEN @i_id2
                        WHEN 3 THEN @i_id3
                        WHEN 4 THEN @i_id4
                        WHEN 5 THEN @i_id5
                        WHEN 6 THEN @i_id6
                        WHEN 7 THEN @i_id7
                        WHEN 8 THEN @i_id8
                        WHEN 9 THEN @i_id9
                        WHEN 10 THEN @i_id10
                        WHEN 11 THEN @i_id11
                        WHEN 12 THEN @i_id12
                        WHEN 13 THEN @i_id13
                        WHEN 14 THEN @i_id14
                        WHEN 15 THEN @i_id15
                    END,

                @li_s_w_id = CASE @li_no
                        WHEN 1 THEN @s_w_id1
                        WHEN 2 THEN @s_w_id2
                        WHEN 3 THEN @s_w_id3
                        WHEN 4 THEN @s_w_id4
                        WHEN 5 THEN @s_w_id5
                        WHEN 6 THEN @s_w_id6
                        WHEN 7 THEN @s_w_id7
                        WHEN 8 THEN @s_w_id8
                        WHEN 9 THEN @s_w_id9
                        WHEN 10 THEN @s_w_id10
                        WHEN 11 THEN @s_w_id11
                        WHEN 12 THEN @s_w_id12
                        WHEN 13 THEN @s_w_id13
                        WHEN 14 THEN @s_w_id14
                        WHEN 15 THEN @s_w_id15
                    END,

                @li_qty = CASE  @li_no
                        WHEN 1 THEN @ol_qty1
                        WHEN 2 THEN @ol_qty2
                        WHEN 3 THEN @ol_qty3
                        WHEN 4 THEN @ol_qty4
                        WHEN 5 THEN @ol_qty5
                        WHEN 6 THEN @ol_qty6
                        WHEN 7 THEN @ol_qty7
                        WHEN 8 THEN @ol_qty8
                        WHEN 9 THEN @ol_qty9
                        WHEN 10 THEN @ol_qty10
                        WHEN 11 THEN @ol_qty11
                        WHEN 12 THEN @ol_qty12
                        WHEN 13 THEN @ol_qty13
                        WHEN 14 THEN @ol_qty14
                        WHEN 15 THEN @ol_qty15
                    END

-----------------------------------------
--  get item data (no one updates item)
-----------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                CASE WHEN (s_quantity - @li_qty < 10) THEN 91
ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1
END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                WHEN 1  THEN s_dist_01
                                WHEN 2  THEN s_dist_02
                                WHEN 3  THEN s_dist_03
                                WHEN 4  THEN s_dist_04
                                WHEN 5  THEN s_dist_05
                                WHEN 6  THEN s_dist_06
                                WHEN 7  THEN s_dist_07
                                WHEN 8  THEN s_dist_08
                                WHEN 9  THEN s_dist_09
                                WHEN 10 THEN s_dist_10
                                END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-----------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-----------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
```

```
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
        SELECT  @i_name,
                @s_quantity,
                b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) >
0) AND

                                (patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
                @i_price,
                @i_price * @li_qty
                                END
                                ELSE
                                BEGIN
-------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition

-------------------------------------------------------------
        SELECT  '',0,'',0,0
        SELECT  @commit_flag    = 0
                                END
             END
-------------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------------
   SELECT  @c_last     = c_last,
           @c_discount = c_discount,
           @c_credit   = c_credit,
           @c_id_local = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_w_id      = @w_id AND
           c_d_id      = @d_id

----------------------------------------
--  insert fresh row into orders table
----------------------------------------
   INSERT INTO orders VALUES ( @o_id,
                           @d_id,
                           @w_id,
                           @c_id_local,
                           0,
                           @o_ol_cnt,
                                @o_entry_d)

-------------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------------
   INSERT INTO new_order VALUES (  @o_id,
                                   @d_id,
                                   @w_id)

------------------------
--  select warehouse tax
------------------------
   SELECT  @w_tax  = w_tax
   FROM    warehouse WITH (repeatableread)
   WHERE   w_id    = @w_id

   IF (@commit_flag = 1)
                                   COMMIT TRANSACTION n
   ELSE
--------------------------------
--  all that work for nuthin!!!
--------------------------------
       ROLLBACK TRANSACTION n

--------------------------------
--  return order data to client
--------------------------------
   SELECT  @w_tax,
           @d_tax,
           @o_id,
           @c_last,
           @c_discount,
           @c_credit,
           @o_entry_d,
           @commit_flag
END
GO
-- End SP31



SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
```

OrdStat.sql

```
------------------------------------------------------------------------
--                                                                    --
--  File:   ORDSTAT.SQL                                               --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                   --
--          Copyright Microsoft, 2005                                 --
--                                                                    --
--          Creates order status stored procedure                     --
--                                                                    --
--          Interface Level:    4.20.000                              --
--                                                                    --
------------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_orderstatus' )
   DROP PROCEDURE tpcc_orderstatus
GO

CREATE PROCEDURE    tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
```

```
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
END
ELSE
BEGIN
    -------------------------------
    --  get customer info if by id
    -------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

------------------------
--  if no such customer
------------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id       = o_id,
        @o_entry_d  = o_entry_d,
        @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id      = @c_id AND
            o_d_id      = @d_id AND
            o_w_id      = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO


-- Create SP0
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp0tpcc_orderstatus' )
    DROP PROCEDURE sp0tpcc_orderstatus
GO

CREATE PROC sp0tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        -------------------------------------------------
        --  get customer id and info using last name
        -------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        -------------------------------
        --  get customer info if by id
        -------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ------------------------
    --  if no such customer
    ------------------------
    IF (@cnt = 0)
    BEGIN
```

```
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
  END

  ------------------
  --  get order info
  ------------------
  SELECT  @o_id           = o_id,
          @o_entry_d      = o_entry_d,
          @o_carrier_id   = o_carrier_id
  FROM    orders WITH (serializable)
  WHERE   o_c_id          = @c_id AND
          o_d_id          = @d_id AND
          o_w_id          = @w_id
  ORDER   BY o_id ASC

  ------------------------------------------------
  --  select order lines for the current order
  ------------------------------------------------
  SELECT  ol_supply_w_id,
          ol_i_id,
          ol_quantity,
          ol_amount,
          ol_delivery_d
  FROM    order_line WITH (repeatableread)
  WHERE   ol_o_id = @o_id AND
          ol_d_id = @d_id AND
          ol_w_id = @w_id

  custnotfound:

  COMMIT TRANSACTION o

  --------------------------
  --  return data to client
  --------------------------
  SELECT  @c_id,
          @c_last,
          @c_first,
          @c_middle,
          @o_entry_d,
          @o_carrier_id,
          @c_balance,
          @o_id
GO

-- End SP0


-- Create SP1
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp1tpcc_orderstatus' )
GODROP PROCEDURE sp1tpcc_orderstatus

CREATE PROC sp1tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   ------------------------
   -- if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ------------------------------------------------
   --  select order lines for the current order
   ------------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   --------------------------
   --  return data to client
   --------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
```

```
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

-- End SP1



-- Create SP2
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp2tpcc_orderstatus' )
   DROP PROCEDURE sp2tpcc_orderstatus
GO

CREATE PROC sp2tpcc_orderstatus
                @w_id       int,
                @d_id       tinyint,

                @c_id       int,
                @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
      @c_first        char(16),
      @c_middle       char(2),
      @o_id           int,
      @o_entry_d      datetime,
      @o_carrier_id   smallint,
      @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_w_id  = @w_id AND
              c_d_id  = @d_id AND
              c_last  = @c_last AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   ------------------------
   --  if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   -------------------
   --  get order info
   -------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ---------------------------------------------
   --  select order lines for the current order
   ---------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id     = @o_id AND
           ol_d_id     = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   --------------------------
   --  return data to client
   --------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO

-- End SP2



-- Create SP3
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp3tpcc_orderstatus' )
   DROP PROCEDURE sp3tpcc_orderstatus
GO

CREATE PROC sp3tpcc_orderstatus
                @w_id       int,
                @d_id       tinyint,

                @c_id       int,
                @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
      @c_first        char(16),
      @c_middle       char(2),
      @o_id           int,
      @o_entry_d      datetime,
      @o_carrier_id   smallint,
      @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
```

```
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
END
ELSE
BEGIN
    --------------------------------
    --  get customer info if by id
    --------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

------------------------
--  if no such customer
------------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id       = o_id,
        @o_entry_d  = o_entry_d,
        @o_carrier_id   = o_carrier_id
    WHERE   o_c_id  = @c_id AND
    FROM    orders WITH (serializable)
            o_d_id      = @d_id AND
            o_w_id      = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP3



-- Create SP4
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp4tpcc_orderstatus' )
    DROP PROCEDURE sp4tpcc_orderstatus
GO

CREATE PROC sp4tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS

DECLARE @c_first    char(16),
        @c_balance  money,
        @c_middle   char(2),
        @o_id       int,
        @o_entry_d  datetime,
        @o_carrier_id   smallint,
        @cnt        smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        --------------------------------
        --  get customer info if by id
        --------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ------------------------
    --  if no such customer
```

```
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id          = o_id,
        @o_entry_d     = o_entry_d,
        @o_carrier_id  = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id         = @c_id AND
        o_d_id         = @d_id AND
        o_w_id         = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
--  select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-------------------------
--  return data to client
-------------------------
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

-- End SP4
```

```
-- Create SP5
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp5tpcc_orderstatus' )
   DROP PROCEDURE sp5tpcc_orderstatus
GO

CREATE PROC sp5tpcc_orderstatus
                  @w_id       int,
                  @d_id       tinyint,

                  @c_id       int,
                  @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   -----------------------
   -- if no such customer
   -----------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id          = o_id,
           @o_entry_d     = o_entry_d,
           @o_carrier_id  = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id         = @c_id AND
           o_d_id         = @d_id AND
           o_w_id         = @w_id
   ORDER   BY o_id ASC

   -----------------------------------------------
   --  select order lines for the current order
   -----------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
```

```
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP5



-- Create SP6
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp6tpcc_orderstatus' )
    DROP PROCEDURE sp6tpcc_orderstatus
GO

CREATE PROC sp6tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_last     = c_last,
                @c_first    = c_first,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        ------------------------------
        --  get customer info if by id
        ------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ------------------------
    --  if no such customer
    ------------------------
    IF (@cnt = 0)
    BEGIN
        RAISERROR('Customer not found',18,1)
        GOTO custnotfound
    END

    ------------------
    --  get order info
    ------------------
    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ---------------------------------------------
    --  select order lines for the current order
    ---------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
        FROM    order_line WITH (repeatableread)
        WHERE   ol_o_id = @o_id AND
                ol_d_id = @d_id AND
                ol_w_id = @w_id

        custnotfound:

        COMMIT TRANSACTION o

        -------------------------
        --  return data to client
        -------------------------
        SELECT  @c_id,
                @c_last,
                @c_first,
                @c_middle,
                @o_entry_d,
                @o_carrier_id,
                @c_balance,
                @o_id
GO

-- End SP6



-- Create SP7
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp7tpcc_orderstatus' )
    DROP PROCEDURE sp7tpcc_orderstatus
GO

CREATE PROC sp7tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
```

```
    ------------------------------------------------
    --  get customer id and info using last name
    ------------------------------------------------
    SELECT  @cnt    = (count(*)+1)/2
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
END
ELSE
BEGIN
    -------------------------------
    --  get customer info if by id
    -------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

------------------------
--  if no such customer
------------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

-------------------
--  get order info

    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP7



-- Create SP8
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp8tpcc_orderstatus' )
    DROP PROCEDURE sp8tpcc_orderstatus
GO

CREATE PROC sp8tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,
                    @c_id       int,
                    @c_last     char(16) = ''
AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        -------------------------------
        --  get customer info if by id
        -------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
```

```
END

-----------------------
-- if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
-- get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
-- select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-------------------------
-- return data to client
-------------------------
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO
```

```
-- End SP8



-- Create SP9
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp9tpcc_orderstatus' )
   DROP PROCEDURE sp9tpcc_orderstatus
GO

CREATE PROC sp9tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      -- get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
```

```
BEGIN
ELSE
   ------------------------------
   -- get customer info if by id
   ------------------------------
   SELECT  @c_balance  = c_balance,
           @c_first    = c_first,
           @c_middle   = c_middle,
           @c_last     = c_last
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_d_id      = @d_id AND
           c_w_id      = @w_id

   SELECT  @cnt          = @@rowcount
END

-----------------------
-- if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
-- get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
-- select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o
```

```
    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP9



-- Create SP10
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp10tpcc_orderstatus' )
    DROP PROCEDURE sp10tpcc_orderstatus
GO

CREATE PROC sp10tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt
        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
END
ELSE
BEGIN
        -------------------------------
        --  get customer info if by id
        -------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------

    --  select order lines for the current order
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP10


-- Create SP11
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp11tpcc_orderstatus' )
    DROP PROCEDURE sp11tpcc_orderstatus
GO

CREATE PROC sp11tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint
```

```
BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      -------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   ------------------------
   --  if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ---------------------------------------------
   --  select order lines for the current order
   ---------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO


-- End SP11



-- Create SP12
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp12tpcc_orderstatus' )
   DROP PROCEDURE sp12tpcc_orderstatus
GO

CREATE PROC sp12tpcc_orderstatus
                @w_id       int,
                @d_id       tinyint,

                @c_id       int,
                @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      -------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
```

```
        c_d_id     = @d_id AND
        c_w_id     = @w_id

   SELECT  @cnt          = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
       @o_entry_d      = o_entry_d,
       @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
       o_d_id          = @d_id AND
       o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
--  select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-------------------------
--  return data to client
-------------------------
SELECT  @c_id,
       @c_last,
       @c_first,
       @c_middle,
       @o_entry_d,
       @c_balance,
       @o_carrier_id,
       @o_id
GO

-- End SP12



-- Create SP13
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp13tpcc_orderstatus' )
   DROP PROCEDURE sp13tpcc_orderstatus
GO

CREATE PROC sp13tpcc_orderstatus
                   @w_id       int,
                   @d_id       tinyint,

                   @c_id       int,
                   @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
       @c_first        char(16),
       @c_middle       char(2),
       @o_id           int,
       @o_entry_d      datetime,
       @o_carrier_id   smallint,
       @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first
      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   -----------------------
   --  if no such customer
   -----------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
          @o_entry_d      = o_entry_d,
          @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
          o_d_id          = @d_id AND
          o_w_id          = @w_id
   ORDER   BY o_id ASC

   -----------------------------------------------
   --  select order lines for the current order
   -----------------------------------------------
   SELECT  ol_supply_w_id,
          ol_i_id,
          ol_quantity,
          ol_amount,
          ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
          ol_d_id = @d_id AND
          ol_w_id = @w_id
```

```
    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP13



-- Create SP14
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp14tpcc_orderstatus' )
    DROP PROCEDURE sp14tpcc_orderstatus
GO

CREATE PROC sp14tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ----------------------------------------------
        --  get customer id and info using last name
        ----------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_w_id  = @w_id AND
                c_last  = @c_last AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
END
ELSE
BEGIN
    -------------------------------
    --  get customer info if by id
    -------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

------------------------
--  if no such customer
------------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

-------------------
--  get order info
-------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
        ORDER BY o_id ASC

    ----------------------------------------------
    --  select order lines for the current order
    ----------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP14



-- Create SP15
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp15tpcc_orderstatus' )
    DROP PROCEDURE sp15tpcc_orderstatus
GO

CREATE PROC sp15tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
```

```
    @o_id           int,
    @o_entry_d      datetime,
    @o_carrier_id   smallint,
    @cnt            smallint

BEGIN TRANSACTION o
  IF (@c_id = 0)
  BEGIN
    ------------------------------------------------
    --  get customer id and info using last name
    ------------------------------------------------
    SELECT  @cnt    = (count(*)+1)/2
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
  END
  ELSE
  BEGIN
    ------------------------------
    --  get customer info if by id
    ------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt          = @@rowcount
  END

  ----------------------
  --  if no such customer
  ----------------------
```

```
  BEGIN
  IF (@cnt = 0)
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
  END

  ------------------
  --  get order info
  ------------------
  SELECT  @o_id           = o_id,
          @o_entry_d      = o_entry_d,
          @o_carrier_id   = o_carrier_id
  FROM    orders WITH (serializable)
  WHERE   o_c_id          = @c_id AND
          o_d_id          = @d_id AND
          o_w_id          = @w_id
  ORDER   BY o_id ASC

  ------------------------------------------------
  --  select order lines for the current order
  ------------------------------------------------
  SELECT  ol_supply_w_id,
          ol_i_id,
          ol_quantity,
          ol_amount,
          ol_delivery_d
  FROM    order_line WITH (repeatableread)
  WHERE   ol_o_id = @o_id AND
          ol_d_id = @d_id AND
          ol_w_id = @w_id

  custnotfound:

  COMMIT TRANSACTION o

  -------------------------
  --  return data to client
  -------------------------
  SELECT  @c_id,
          @c_last,
          @c_first,
          @c_middle,
          @o_entry_d,
          @o_carrier_id,
          @c_balance,
          @o_id
GO

-- End SP15



-- Create SP16
```

```
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp16tpcc_orderstatus' )  DROP PROCEDURE sp16tpcc_orderstatus
GO

CREATE PROC sp16tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
  IF (@c_id = 0)
  BEGIN
    ------------------------------------------------
    --  get customer id and info using last name
    ------------------------------------------------
    SELECT  @cnt    = (count(*)+1)/2
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
  END
  ELSE
  BEGIN
    ------------------------------
    --  get customer info if by id
    ------------------------------
    SELECT  @c_balance  = c_balance,
```

```
        @c_first    = c_first,
        @c_middle   = c_middle,
        @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
        c_d_id      = @d_id AND
        c_w_id      = @w_id

    SELECT  @cnt          = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
--  select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-------------------------
--  return data to client
-------------------------
    SELECT  @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP16



-- Create SP17
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp17tpcc_orderstatus' )
    DROP PROCEDURE sp17tpcc_orderstatus
GO

CREATE PROC sp17tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        -----------------------------------------------
        --  get customer id and info using last name
        -----------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        -----------------------------
        --  get customer info if by id
        -----------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt          = @@rowcount
    END

    -----------------------
    --  if no such customer
    -----------------------
    IF (@cnt = 0)
    BEGIN
        RAISERROR('Customer not found',18,1)
        GOTO custnotfound
    END

    ------------------
    --  get order info
    ------------------
    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    -----------------------------------------------
    --  select order lines for the current order
    -----------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
```

```
        ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP17



-- Create SP18
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp18tpcc_orderstatus' )
    DROP PROCEDURE sp18tpcc_orderstatus
GO

CREATE PROC sp18tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
    -- get customer id and info using last name
    ------------------------------------------
    SELECT  @cnt    = (count(*)+1)/2
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
END
ELSE
BEGIN
    ------------------------------
    --  get customer info if by id
    ------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id       = o_id,
        @o_carrier_id   = o_carrier_id,
        @o_entry_d      = o_entry_d
    FROM    orders WITH (serializable)
    WHERE   o_c_id      = @c_id AND
            o_d_id      = @d_id AND
            o_w_id      = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP18



-- Create SP19
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp19tpcc_orderstatus' )
    DROP PROCEDURE sp19tpcc_orderstatus
GO

CREATE PROC sp19tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''
```

```
AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      -------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt        = @@rowcount

   END
   ------------------------
   --  if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ------------------------------------------------
   --  select order lines for the current order
   ------------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO

-- End SP19

-- Create SP20
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp20tpcc_orderstatus' )
   DROP PROCEDURE sp20tpcc_orderstatus
GO

CREATE PROC sp20tpcc_orderstatus
                   @w_id       int,
                   @d_id       tinyint,

                   @c_id       int,
                   @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
```

```
   ------------------------------
   --  get customer info if by id
   ------------------------------
   SELECT  @c_balance  = c_balance,
           @c_first    = c_first,
           @c_middle   = c_middle,
           @c_last     = c_last
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_d_id      = @d_id AND
           c_w_id      = @w_id

   SELECT  @cnt          = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

----------------------------------------------
--  select order lines for the current order
----------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o
```

```
   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO

-- End SP20

-- Create SP21
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp21tpcc_orderstatus' )
   DROP PROCEDURE sp21tpcc_orderstatus
GO

CREATE PROC sp21tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ----------------------------------------------
      --  get customer id and info using last name
      ----------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_last     = c_last,
              @c_first    = c_first,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   -----------------------
   --  if no such customer
   -----------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ----------------------------------------------
   --  select order lines for the current order
   ----------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
```

```
        ol_quantity,
        ol_amount,
        ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
        ol_d_id = @d_id AND
        ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

-- End SP21



-- Create SP22
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp22tpcc_orderstatus' )
    DROP PROCEDURE sp22tpcc_orderstatus
GO

CREATE PROC sp22tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o

BEGIN
IF (@c_id = 0)
    ----------------------------------------------
    --  get customer id and info using last name
    ----------------------------------------------
    SELECT  @cnt    = (count(*)+1)/2
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
            c_w_id  = @w_id AND
            c_d_id  = @d_id

    SET     rowcount @cnt

    SELECT  @c_id       = c_id,
            @c_balance  = c_balance,
            @c_first    = c_first,
            @c_last     = c_last,
            @c_middle   = c_middle
    FROM    customer WITH (repeatableread)
    WHERE   c_last      = @c_last AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id
    ORDER   BY c_w_id, c_d_id, c_last, c_first

    SET rowcount 0
END
ELSE
BEGIN
    ------------------------------
    --  get customer info if by id
    ------------------------------
    SELECT  @c_balance  = c_balance,
            @c_first    = c_first,
            @c_middle   = c_middle,
            @c_last     = c_last
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_d_id      = @d_id AND
            c_w_id      = @w_id

    SELECT  @cnt        = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

------------------
--  get order info

    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP22



-- Create SP23
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp23tpcc_orderstatus' )
    DROP PROCEDURE sp23tpcc_orderstatus
GO

CREATE PROC sp23tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,
```

```
                @c_id       int,
                @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      -------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt        = @@rowcount
   END

   ------------------------
   --  if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id          = o_id,
           @o_entry_d     = o_entry_d,
           @o_carrier_id  = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id         = @c_id AND
           o_d_id         = @d_id AND
           o_w_id         = @w_id
   ORDER   BY o_id ASC

   ------------------------------------------------
   --  select order lines for the current order
   ------------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO
```

```
-- End SP23


-- Create SP24
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp24tpcc_orderstatus' )
   DROP PROCEDURE sp24tpcc_orderstatus
GO

CREATE PROC sp24tpcc_orderstatus
                @w_id       int,
                @d_id       tinyint,

                @c_id       int,
                @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
```

```
ELSE
BEGIN
   ------------------------------
   --  get customer info if by id
   ------------------------------
   SELECT  @c_balance  = c_balance,
          @c_first    = c_first,
          @c_middle   = c_middle,
          @c_last     = c_last
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
          c_d_id      = @d_id AND
          c_w_id      = @w_id

   SELECT  @cnt          = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
       @o_entry_d      = o_entry_d,
       @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
       o_d_id          = @d_id AND
       o_w_id          = @w_id
ORDER   BY o_id ASC

-----------------------------------------------
--  select order lines for the current order
-----------------------------------------------
SELECT  ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
          @c_last,
          @c_first,
          @c_middle,
          @o_entry_d,
          @o_carrier_id,
          @c_balance,
          @o_id
GO

-- End SP24



-- Create SP25
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp25tpcc_orderstatus' )
   DROP PROCEDURE sp25tpcc_orderstatus
GO

CREATE PROC sp25tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
       @c_first        char(16),
       @c_middle       char(2),
       @o_id           int,
       @o_entry_d      datetime,
       @o_carrier_id   smallint,
       @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ----------------------------------------------
      --  get customer id and info using last name
      ----------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
             c_w_id  = @w_id AND
             c_d_id  = @d_id

      SET     rowcount @cnt
      SELECT  @c_id       = c_id,
             @c_balance  = c_balance,
             @c_first    = c_first,
             @c_last     = c_last,
             @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
             c_w_id      = @w_id AND
             c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
             @c_first    = c_first,
             @c_middle   = c_middle,
             @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
             c_d_id      = @d_id AND
             c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   -----------------------
   --  if no such customer
   -----------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
          @o_entry_d      = o_entry_d,
          @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
          o_d_id          = @d_id AND
          o_w_id          = @w_id
   ORDER   BY o_id ASC

   -----------------------------------------------
```

```
    -- select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP25



-- Create SP26
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp26tpcc_orderstatus' )
    DROP PROCEDURE sp26tpcc_orderstatus
GO

CREATE PROC sp26tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @cnt            smallint,
        @o_carrier_id   smallint,

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        ------------------------------
        --  get customer info if by id
        ------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ------------------------
    --  if no such customer
    ------------------------
    IF (@cnt = 0)
    BEGIN
        RAISERROR('Customer not found',18,1)
        GOTO custnotfound
    END
    ------------------
    --  get order info
    ------------------
    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP26



-- Create SP27
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp27tpcc_orderstatus' )
    DROP PROCEDURE sp27tpcc_orderstatus
GO
```

```
CREATE PROC sp27tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      -------------------------------
      --  get customer info if by id
      -------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   ------------------------
   --  if no such customer
   ------------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
           o_w_id          = @w_id
   ORDER   BY o_id ASC

   ------------------------------------------------
   --  select order lines for the current order
   ------------------------------------------------
   SELECT  ol_supply_w_id,
           ol_i_id,
           ol_quantity,
           ol_amount,
           ol_delivery_d
   FROM    order_line WITH (repeatableread)
   WHERE   ol_o_id = @o_id AND
           ol_d_id = @d_id AND
           ol_w_id = @w_id

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @c_balance,
           @o_carrier_id,
           @o_id
GO

-- End SP27



-- Create SP28
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp28tpcc_orderstatus' )
   DROP PROCEDURE sp28tpcc_orderstatus
GO

CREATE PROC sp28tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ------------------------------------------------
      --  get customer id and info using last name
      ------------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
              c_w_id  = @w_id AND
              c_d_id  = @d_id

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
```

```
   ORDER   BY c_w_id, c_d_id, c_last, c_first

   SET rowcount 0
END
ELSE
BEGIN
   ------------------------------
   --  get customer info if by id
   ------------------------------
   SELECT  @c_balance  = c_balance,
           @c_first    = c_first,
           @c_middle   = c_middle,
           @c_last     = c_last
   FROM    customer WITH (repeatableread)
   WHERE   c_id        = @c_id AND
           c_d_id      = @d_id AND
           c_w_id      = @w_id

   SELECT  @cnt          = @@rowcount
END

-----------------------
--  if no such customer
-----------------------
IF (@cnt = 0)
BEGIN
   RAISERROR('Customer not found',18,1)
   GOTO custnotfound
END

------------------
--  get order info
------------------
SELECT  @o_id           = o_id,
        @o_entry_d      = o_entry_d,
        @o_carrier_id   = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id          = @c_id AND
        o_d_id          = @d_id AND
        o_w_id          = @w_id
ORDER   BY o_id ASC

---------------------------------------------
--  select order lines for the current order
---------------------------------------------
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
        ol_w_id = @w_id AND
        ol_d_id = @d_id AND

   custnotfound:

   COMMIT TRANSACTION o

   -------------------------
   --  return data to client
   -------------------------
   SELECT  @c_id,
           @c_last,
           @c_first,
           @c_middle,
           @o_entry_d,
           @o_carrier_id,
           @c_balance,
           @o_id
GO

-- End SP28



-- Create SP29
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp29tpcc_orderstatus' )
   DROP PROCEDURE sp29tpcc_orderstatus
GO

CREATE PROC sp29tpcc_orderstatus
                   @w_id       int,
                   @d_id       tinyint,

                   @c_id       int,
                   @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
   IF (@c_id = 0)
   BEGIN
      ---------------------------------------------
      --  get customer id and info using last name
      ---------------------------------------------
      SELECT  @cnt    = (count(*)+1)/2
      FROM    customer WITH (repeatableread)
      WHERE   c_w_id  = @w_id AND
              c_d_id  = @d_id AND
              c_last  = @c_last

      SET     rowcount @cnt

      SELECT  @c_id       = c_id,
              @c_balance  = c_balance,
              @c_first    = c_first,
              @c_last     = c_last,
              @c_middle   = c_middle
      FROM    customer WITH (repeatableread)
      WHERE   c_last      = @c_last AND
              c_w_id      = @w_id AND
              c_d_id      = @d_id
      ORDER   BY c_w_id, c_d_id, c_last, c_first

      SET rowcount 0
   END
   ELSE
   BEGIN
      ------------------------------
      --  get customer info if by id
      ------------------------------
      SELECT  @c_balance  = c_balance,
              @c_first    = c_first,
              @c_middle   = c_middle,
              @c_last     = c_last
      FROM    customer WITH (repeatableread)
      WHERE   c_id        = @c_id AND
              c_d_id      = @d_id AND
              c_w_id      = @w_id

      SELECT  @cnt          = @@rowcount
   END

   -----------------------
   --  if no such customer
   -----------------------
   IF (@cnt = 0)
   BEGIN
      RAISERROR('Customer not found',18,1)
      GOTO custnotfound
   END

   ------------------
   --  get order info
   ------------------
   SELECT  @o_id           = o_id,
           @o_entry_d      = o_entry_d,
           @o_carrier_id   = o_carrier_id
   FROM    orders WITH (serializable)
   WHERE   o_c_id          = @c_id AND
           o_d_id          = @d_id AND
```

```
        o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    --------------------------
    --  return data to client
    --------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP29



-- Create SP30
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp30tpcc_orderstatus' )
    DROP PROCEDURE sp30tpcc_orderstatus
GO

CREATE PROC sp30tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_middle       char(2),
        @c_first        char(16),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        ------------------------------
        --  get customer info if by id
        ------------------------------
        SELECT  @c_balance  = c_balance,
                @c_first    = c_first,
                @c_middle   = c_middle,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ------------------------
    --  if no such customer
    ------------------------
    BEGIN
        IF (@cnt = 0)
            RAISERROR('Customer not found',18,1)
            GOTO custnotfound
    END

    ------------------
    --  get order info
    ------------------
    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    --------------------------
    --  return data to client
    --------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP30


-- Create SP31
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp31tpcc_orderstatus' )
```

```
    DROP PROCEDURE sp31tpcc_orderstatus
GO

CREATE PROC sp31tpcc_orderstatus
                    @w_id       int,
                    @d_id       tinyint,

                    @c_id       int,
                    @c_last     char(16) = ''

AS
DECLARE @c_balance      money,
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

BEGIN TRANSACTION o
    IF (@c_id = 0)
    BEGIN
        ------------------------------------------------
        --  get customer id and info using last name
        ------------------------------------------------
        SELECT  @cnt    = (count(*)+1)/2
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                c_w_id  = @w_id AND
                c_d_id  = @d_id

        SET     rowcount @cnt

        SELECT  @c_id       = c_id,
                @c_balance  = c_balance,
                @c_first    = c_first,
                @c_last     = c_last,
                @c_middle   = c_middle
        FROM    customer WITH (repeatableread)
        WHERE   c_last      = @c_last AND
                c_w_id      = @w_id AND
                c_d_id      = @d_id
        ORDER   BY c_w_id, c_d_id, c_last, c_first

        SET rowcount 0
    END
    ELSE
    BEGIN
        ------------------------------
        --  get customer info if by id
        ------------------------------
        SELECT  @c_balance  = c_balance,
                @c_middle   = c_middle,
                @c_first    = c_first,
                @c_last     = c_last
        FROM    customer WITH (repeatableread)
        WHERE   c_id        = @c_id AND
                c_d_id      = @d_id AND
                c_w_id      = @w_id

        SELECT  @cnt        = @@rowcount
    END

    ----------------------
    --  if no such customer
    ----------------------
    IF (@cnt = 0)
    BEGIN
        RAISERROR('Customer not found',18,1)
        GOTO custnotfound
    END

    ------------------
    --  get order info
    ------------------
    SELECT  @o_id           = o_id,
            @o_entry_d      = o_entry_d,
            @o_carrier_id   = o_carrier_id
    FROM    orders WITH (serializable)
    WHERE   o_c_id          = @c_id AND
            o_d_id          = @d_id AND
            o_w_id          = @w_id
    ORDER   BY o_id ASC

    ------------------------------------------------
    --  select order lines for the current order
    ------------------------------------------------
    SELECT  ol_supply_w_id,
            ol_i_id,
            ol_quantity,
            ol_amount,
            ol_delivery_d
    FROM    order_line WITH (repeatableread)
    WHERE   ol_o_id = @o_id AND
            ol_d_id = @d_id AND
            ol_w_id = @w_id

    custnotfound:

    COMMIT TRANSACTION o

    -------------------------
    --  return data to client
    -------------------------
    SELECT  @c_id,
            @c_last,
            @c_first,
            @c_middle,
            @o_entry_d,
            @o_carrier_id,
            @c_balance,
            @o_id
GO

-- End SP31
```

delivery.sql

```
------------------------------------------------------------------------
--                                                                    --
--  File:   DELIVERY.SQL                                              --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                   --
--          Copyright Microsoft, 2005                                 --
--                                                                    --
--          Creates 32 delivery stored procedures                     --
--                                                                    --
--          Interface Level:    4.20.000                              --
--                                                                    --
------------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_delivery' )
    DROP PROCEDURE tpcc_delivery
GO

CREATE PROC tpcc_delivery
                @w_id           int,

                @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
```

```
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
            @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
            @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
            @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
            @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
            @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
            @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
            @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
            @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
            @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- Create SP0
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp0tpcc_delivery' )
   DROP PROCEDURE sp0tpcc_delivery
GO

CREATE PROC sp0tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
```

```
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP0


-- Create SP1
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp1tpcc_delivery' )
   DROP PROCEDURE sp1tpcc_delivery
GO

CREATE PROC sp1tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint
AS
DECLARE @d_id       tinyint,
       @o_id       int,
       @c_id       int,
       @total      money,
       @oid1       int,
       @oid2       int,
       @oid3       int,
       @oid4       int,
       @oid5       int,
       @oid6       int,
       @oid7       int,
       @oid8       int,
       @oid9       int,
       @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP1



-- Create SP2
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp2tpcc_delivery' )
   DROP PROCEDURE sp2tpcc_delivery
GO

CREATE PROC sp2tpcc_delivery
            @w_id           int,
```

```
        @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
              @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
              @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
              @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
              @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
              @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
              @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
              @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
              @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
              @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP2



-- Create SP3
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp3tpcc_delivery' )
    DROP PROCEDURE sp3tpcc_delivery
GO

CREATE PROC sp3tpcc_delivery
        @w_id           int,

        @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id
```

```
        -- set date in all lineitems for this order (and sum amounts)
        UPDATE  order_line
        SET     ol_delivery_d   = GETDATE(),
                @total          = @total + ol_amount
        WHERE   ol_w_id         = @w_id AND
                ol_d_id         = @d_id AND
                ol_o_id         = @o_id

        -- accummulate lineitem amounts for this order into customer
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP3



-- Create SP4

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp4tpcc_delivery' )
    DROP PROCEDURE sp4tpcc_delivery
GO

CREATE PROC sp4tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id = @o_carrier_id,
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
              @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
              @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
              @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
              @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
              @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
              @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
              @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
              @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
              @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP4
```

```
-- Create SP5
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp5tpcc_delivery' )
   DROP PROCEDURE sp5tpcc_delivery
GO

CREATE PROC sp5tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id
         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
         @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
         @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
         @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
         @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
         @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
         @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
         @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
         @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
         @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO
-- End SP5



-- Create SP6
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp6tpcc_delivery' )
   DROP PROCEDURE sp6tpcc_delivery
GO

CREATE PROC sp6tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
```

```
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id AND
                no_o_id = @o_id

        -- set carrier_id on this order (and get customer id)
        UPDATE  orders
        SET     o_carrier_id    = @o_carrier_id,
                @c_id           = o_c_id
        WHERE   o_w_id          = @w_id AND
                o_d_id          = @d_id AND
                o_id            = @o_id

        -- set date in all lineitems for this order (and sum amounts)
        UPDATE  order_line
        SET     ol_delivery_d   = GETDATE(),
                @total          = @total + ol_amount
        WHERE   ol_w_id         = @w_id AND
                ol_d_id         = @d_id AND
                ol_o_id         = @o_id

        -- accummulate lineitem amounts for this order into customer
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP6



-- Create SP7
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp7tpcc_delivery' )
   DROP PROCEDURE sp7tpcc_delivery
GO

CREATE PROC sp7tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      BEGIN
      IF (@@rowcount <> 0)
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
       END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
```

```
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP7



-- Create SP8
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp8tpcc_delivery' )
    DROP PROCEDURE sp8tpcc_delivery
GO

CREATE PROC sp8tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_d_id = @d_id AND
              no_w_id = @w_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
              @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
              @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
              @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
              @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
              @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
              @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
              @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
              @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
              @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
      END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP8



-- Create SP9
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp9tpcc_delivery' )
    DROP PROCEDURE sp9tpcc_delivery
GO

CREATE PROC sp9tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
```

```
        @o_id   = no_o_id
    FROM    new_order WITH  (serializable updlock)
    WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
    ORDER   BY no_o_id ASC

    IF (@@rowcount <> 0)
    BEGIN
       -- claim the order for this district
       DELETE  new_order
       WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

       -- set carrier_id on this order (and get customer id)
       UPDATE  orders
       SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
       WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

       -- set date in all lineitems for this order (and sum amounts)
       UPDATE  order_line
       SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
       WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

       -- accummulate lineitem amounts for this order into customer
       UPDATE  customer
       SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
       WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d
-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP9



-- Create SP10
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp10tpcc_delivery' )
    DROP PROCEDURE sp10tpcc_delivery
GO

CREATE PROC sp10tpcc_delivery
                @w_id           int,

                @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
    WHILE (@d_id < 10)
    BEGIN
        SELECT  @d_id   = @d_id + 1,
                @total  = 0,
        SELECT  TOP 1
                @o_id   = no_o_id
        FROM    new_order WITH  (serializable updlock)
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id
        ORDER   BY no_o_id ASC

        IF (@@rowcount <> 0)
        BEGIN
           -- claim the order for this district
           DELETE  new_order
           WHERE   no_w_id = @w_id AND
                   no_d_id = @d_id AND
                   no_o_id = @o_id

           -- set carrier_id on this order (and get customer id)
           UPDATE  orders
           SET     o_carrier_id    = @o_carrier_id,
                   @c_id           = o_c_id
           WHERE   o_w_id          = @w_id AND
                   o_d_id          = @d_id AND
                   o_id            = @o_id

           -- set date in all lineitems for this order (and sum amounts)
           UPDATE  order_line
           SET     ol_delivery_d   = GETDATE(),
                   @total          = @total + ol_amount
           WHERE   ol_w_id         = @w_id AND
                   ol_d_id         = @d_id AND
                   ol_o_id         = @o_id

           -- accummulate lineitem amounts for this order into customer
           UPDATE  customer
           SET     c_balance       = c_balance + @total,
                   c_delivery_cnt  = c_delivery_cnt + 1
           WHERE   c_w_id          = @w_id AND
                   c_d_id          = @d_id AND
                   c_id            = @c_id
        END

       SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
```

```
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP10


-- Create SP11
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp11tpcc_delivery' )
   DROP PROCEDURE sp11tpcc_delivery
GO

CREATE PROC sp11tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d

BEGIN
WHILE (@d_id < 10)
   SELECT  @d_id   = @d_id + 1,
           @total  = 0,
           @o_id   = 0

   SELECT  TOP 1
           @o_id   = no_o_id
   FROM    new_order WITH  (serializable updlock)
   WHERE   no_w_id = @w_id AND
           no_d_id = @d_id
   ORDER   BY no_o_id ASC

   IF (@@rowcount <> 0)
   BEGIN
      -- claim the order for this district
      DELETE  new_order
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id AND
              no_o_id = @o_id

      -- set carrier_id on this order (and get customer id)
      UPDATE  orders
      SET     o_carrier_id    = @o_carrier_id,
              @c_id           = o_c_id
      WHERE   o_w_id          = @w_id AND
              o_d_id          = @d_id AND
              o_id            = @o_id

      -- set date in all lineitems for this order (and sum amounts)
      UPDATE  order_line
      SET     ol_delivery_d   = GETDATE(),
              @total          = @total + ol_amount
      WHERE   ol_w_id         = @w_id AND
              ol_d_id         = @d_id AND
              ol_o_id         = @o_id

      -- accummulate lineitem amounts for this order into customer
      UPDATE  customer
      SET     c_balance       = c_balance + @total,
              c_delivery_cnt  = c_delivery_cnt + 1
      WHERE   c_w_id          = @w_id AND
              c_d_id          = @d_id AND
              c_id            = @c_id
   END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP11



-- Create SP12
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp12tpcc_delivery' )
   DROP PROCEDURE sp12tpcc_delivery
GO

CREATE PROC sp12tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0
```

```
BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
         @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
         @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
         @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
         @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
         @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
         @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
         @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
         @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
         @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP12



-- Create SP13
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp13tpcc_delivery' )
   DROP PROCEDURE sp13tpcc_delivery
GO

CREATE PROC sp13tpcc_delivery
            @w_id          int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int;

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
```

```
        @oid2   = CASE @d_id WHEN  2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN  3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN  4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN  5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN  6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN  7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN  8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN  9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP13



-- Create SP14
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp14tpcc_delivery' )
    DROP PROCEDURE sp14tpcc_delivery
GO

CREATE PROC sp14tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
    WHILE (@d_id < 10)
    BEGIN
        SELECT  @d_id   = @d_id + 1,
                @total  = 0,
                @o_id   = 0

        SELECT  TOP 1
                @o_id   = no_o_id
        FROM    new_order WITH  (serializable updlock)
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id
        ORDER   BY no_o_id ASC

        IF (@@rowcount <> 0)
        BEGIN
            --  claim the order for this district
            DELETE  new_order
            WHERE   no_w_id = @w_id AND
                    no_d_id = @d_id AND
                    no_o_id = @o_id

            --  set carrier_id on this order (and get customer id)
            UPDATE  orders
            SET     o_carrier_id    = @o_carrier_id,
                    @c_id           = o_c_id
            WHERE   o_w_id          = @w_id AND
                    o_d_id          = @d_id AND
                    o_id            = @o_id

            --  set date in all lineitems for this order (and sum amounts)
            UPDATE  order_line
            SET     ol_delivery_d   = GETDATE(),
                    @total          = @total + ol_amount
            WHERE   ol_w_id         = @w_id AND
                    ol_d_id         = @d_id AND
                    ol_o_id         = @o_id

            --  accummulate lineitem amounts for this order into customer
            UPDATE  customer
            SET     c_balance       = c_balance + @total,
                    c_delivery_cnt  = c_delivery_cnt + 1
            WHERE   c_w_id          = @w_id AND
                    c_d_id          = @d_id AND
                    c_id            = @c_id
        END
    SELECT  @oid1   = CASE @d_id WHEN  1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN  2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN  3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN  4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN  5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN  6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN  7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN  8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN  9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP14



-- Create SP15
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp15tpcc_delivery' )
    DROP PROCEDURE sp15tpcc_delivery
GO

CREATE PROC sp15tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
```

```
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
   END,
            @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
            @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
            @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
            @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
            @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
            @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
            @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
            @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
            @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
      END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP15



-- Create SP16
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp16tpcc_delivery' )
   DROP PROCEDURE sp16tpcc_delivery
GO

CREATE PROC sp16tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @total      money,
      @c_id       int,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
```

```
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP16



-- Create SP17
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp17tpcc_delivery' )
   DROP PROCEDURE sp17tpcc_delivery
GO

CREATE PROC sp17tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint
AS
DECLARE @d_id       tinyint,
       @o_id       int,
       @c_id       int,
       @total      money,
       @oid1       int,
       @oid2       int,
       @oid3       int,
       @oid4       int,
       @oid5       int,
       @oid6       int,
       @oid7       int,
       @oid8       int,
       @oid9       int,
       @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP17



-- Create SP18
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp18tpcc_delivery' )
   DROP PROCEDURE sp18tpcc_delivery
GO

CREATE PROC sp18tpcc_delivery
```

```
        @w_id           int,

        @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
         @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
         @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
         @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
         @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
         @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
         @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
         @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
         @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
         @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP18



-- Create SP19
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp19tpcc_delivery' )
GODROP PROCEDURE sp19tpcc_delivery

CREATE PROC sp19tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
```

```
            o_d_id          = @d_id AND
            o_id            = @o_id

        -- set date in all lineitems for this order (and sum amounts)
        UPDATE  order_line
        SET     ol_delivery_d   = GETDATE(),
                @total          = @total + ol_amount
        WHERE   ol_w_id         = @w_id AND
                ol_d_id         = @d_id AND
                ol_o_id         = @o_id

        -- accummulate lineitem amounts for this order into customer
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP19
```

```
-- Create SP20
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp20tpcc_delivery' )
   DROP PROCEDURE sp20tpcc_delivery
GO

CREATE PROC sp20tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP20
```

```
-- Create SP21
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp21tpcc_delivery' )
   DROP PROCEDURE sp21tpcc_delivery
GO

CREATE PROC sp21tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
         @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
         @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
         @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
         @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
         @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
         @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
         @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
         @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
         @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP21



-- Create SP22
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp22tpcc_delivery' )
   DROP PROCEDURE sp22tpcc_delivery
GO

CREATE PROC sp22tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
```

```
            no_d_id = @d_id AND
            no_o_id = @o_id

        -- set carrier_id on this order (and get customer id)
        UPDATE  orders
        SET     o_carrier_id    = @o_carrier_id,
                @c_id           = o_c_id
        WHERE   o_w_id          = @w_id AND
                o_d_id          = @d_id AND
                o_id            = @o_id

        -- set date in all lineitems for this order (and sum amounts)
        UPDATE  order_line
        SET     ol_delivery_d   = GETDATE(),
                @total          = @total + ol_amount
        WHERE   ol_w_id         = @w_id AND
                ol_d_id         = @d_id AND
                ol_o_id         = @o_id

        -- accummulate lineitem amounts for this order into customer
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
      @oid6,
      @oid7,
      @oid8,
      @oid9,
      @oid10
GO

-- End SP22



-- Create SP23
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp23tpcc_delivery' )
   DROP PROCEDURE sp23tpcc_delivery
GO

CREATE PROC sp23tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
      @o_id       int,
      @c_id       int,
      @total      money,
      @oid1       int,
      @oid2       int,
      @oid3       int,
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

      SELECT  TOP 1
            @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
            @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
            @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
            @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
            @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
            @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
            @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
            @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
            @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
            @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
      END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,
```

```
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP23



-- Create SP24
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp24tpcc_delivery' )
    DROP PROCEDURE sp24tpcc_delivery
GO

CREATE PROC sp24tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         -- claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         -- set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         -- set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         -- accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

      SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
              @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
              @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
              @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
              @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
              @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
              @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
              @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
              @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
              @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP24



-- Create SP25
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp25tpcc_delivery' )
    DROP PROCEDURE sp25tpcc_delivery
GO

CREATE PROC sp25tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
```

```
    FROM    new_order WITH  (serializable updlock)
    WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
    ORDER   BY no_o_id ASC

    IF (@@rowcount <> 0)
    BEGIN
       --  claim the order for this district
       DELETE  new_order
       WHERE   no_w_id = @w_id AND
               no_d_id = @d_id AND
               no_o_id = @o_id

       --  set carrier_id on this order (and get customer id)
       UPDATE  orders
       SET     o_carrier_id    = @o_carrier_id,
               @c_id           = o_c_id
       WHERE   o_w_id          = @w_id AND
               o_d_id          = @d_id AND
               o_id            = @o_id

       --  set date in all lineitems for this order (and sum amounts)
       UPDATE  order_line
       SET     ol_delivery_d   = GETDATE(),
               @total          = @total + ol_amount
       WHERE   ol_w_id         = @w_id AND
               ol_d_id         = @d_id AND
               ol_o_id         = @o_id

       --  accummulate lineitem amounts for this order into customer
       UPDATE  customer
       SET     c_balance       = c_balance + @total,
               c_delivery_cnt  = c_delivery_cnt + 1
       WHERE   c_w_id          = @w_id AND
               c_d_id          = @d_id AND
               c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
       @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
       @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
       @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
       @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
       @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
       @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
       @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
       @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
       @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
GO

-- End SP25



-- Create SP26
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp26tpcc_delivery' )
   DROP PROCEDURE sp26tpcc_delivery
GO

CREATE PROC sp26tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
       @o_id       int,
       @c_id       int,
       @total      money,
       @oid1       int,
       @oid2       int,
       @oid3       int,
       @oid4       int,
       @oid5       int,
       @oid6       int,
       @oid7       int,
       @oid8       int,
       @oid9       int,
       @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0
      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

     SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
       @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
       @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
       @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
       @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
       @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
       @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
       @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
       @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
       @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END
```

```
COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP26



-- Create SP27
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp27tpcc_delivery' )
    DROP PROCEDURE sp27tpcc_delivery
GO

CREATE PROC sp27tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d

BEGIN
WHILE (@d_id < 10)
    SELECT  @d_id   = @d_id + 1,
            @total  = 0,
            @o_id   = 0

    SELECT  TOP 1
            @o_id   = no_o_id
    FROM    new_order WITH  (serializable updlock)
    WHERE   no_w_id = @w_id AND
            no_d_id = @d_id
    ORDER   BY no_o_id ASC

    IF (@@rowcount <> 0)
    BEGIN
        --  claim the order for this district
        DELETE  new_order
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id AND
                no_o_id = @o_id

        --  set carrier_id on this order (and get customer id)
        UPDATE  orders
        SET     o_carrier_id    = @o_carrier_id,
                @c_id           = o_c_id
        WHERE   o_w_id          = @w_id AND
                o_d_id          = @d_id AND
                o_id            = @o_id

        --  set date in all lineitems for this order (and sum amounts)
        UPDATE  order_line
        SET     ol_delivery_d   = GETDATE(),
                @total          = @total + ol_amount
        WHERE   ol_w_id         = @w_id AND
                ol_d_id         = @d_id AND
                ol_o_id         = @o_id

        --  accummulate lineitem amounts for this order into customer
        UPDATE  customer
        SET     c_balance       = c_balance + @total,
                c_delivery_cnt  = c_delivery_cnt + 1
        WHERE   c_w_id          = @w_id AND
                c_d_id          = @d_id AND
                c_id            = @c_id
    END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP27



-- Create SP28
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp28tpcc_delivery' )
    DROP PROCEDURE sp28tpcc_delivery
GO

CREATE PROC sp28tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0
```

```
BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP28



-- Create SP29
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp29tpcc_delivery' )
   DROP PROCEDURE sp29tpcc_delivery
GO

CREATE PROC sp29tpcc_delivery
             @w_id          int,

             @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int;

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_w_id          = @w_id AND
                 c_d_id          = @d_id AND
                 c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
```

```
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP29



-- Create SP30
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp30tpcc_delivery' )
    DROP PROCEDURE sp30tpcc_delivery
GO

CREATE PROC sp30tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
        @oid4       int,
        @oid5       int,
        @oid6       int,
        @oid7       int,
        @oid8       int,
        @oid9       int,
        @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
    WHILE (@d_id < 10)
    BEGIN
        SELECT  @d_id   = @d_id + 1,
                @total  = 0,
                @o_id   = 0

        SELECT  TOP 1
                @o_id   = no_o_id
        FROM    new_order WITH  (serializable updlock)
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id
        ORDER   BY no_o_id ASC

        IF (@@rowcount <> 0)
        BEGIN
            --  claim the order for this district
            DELETE  new_order
            WHERE   no_w_id = @w_id AND
                    no_d_id = @d_id AND
                    no_o_id = @o_id

            --  set carrier_id on this order (and get customer id)
            UPDATE  orders
            SET     o_carrier_id    = @o_carrier_id,
                    @c_id           = o_c_id
            WHERE   o_w_id          = @w_id AND
                    o_d_id          = @d_id AND
                    o_id            = @o_id

            --  set date in all lineitems for this order (and sum amounts)
            UPDATE  order_line
            SET     ol_delivery_d   = GETDATE(),
                    @total          = @total + ol_amount
            WHERE   ol_w_id         = @w_id AND
                    ol_d_id         = @d_id AND
                    ol_o_id         = @o_id

            --  accummulate lineitem amounts for this order into customer
            UPDATE  customer
            SET     c_balance       = c_balance + @total,
                    c_delivery_cnt  = c_delivery_cnt + 1
            WHERE   c_w_id          = @w_id AND
                    c_d_id          = @d_id AND
                    c_id            = @c_id
        END
    SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
        @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
        @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
        @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
        @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
        @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
        @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
        @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
        @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
        @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP30


-- Create SP31
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp31tpcc_delivery' )
    DROP PROCEDURE sp31tpcc_delivery
GO

CREATE PROC sp31tpcc_delivery
            @w_id           int,

            @o_carrier_id   smallint

AS

DECLARE @d_id       tinyint,
        @o_id       int,
        @c_id       int,
        @total      money,
        @oid1       int,
        @oid2       int,
        @oid3       int,
```

```
      @oid4       int,
      @oid5       int,
      @oid6       int,
      @oid7       int,
      @oid8       int,
      @oid9       int,
      @oid10      int

SELECT  @d_id   = 0

BEGIN TRANSACTION d
   WHILE (@d_id < 10)
   BEGIN
      SELECT  @d_id   = @d_id + 1,
              @total  = 0,
              @o_id   = 0

      SELECT  TOP 1
              @o_id   = no_o_id
      FROM    new_order WITH  (serializable updlock)
      WHERE   no_w_id = @w_id AND
              no_d_id = @d_id
      ORDER   BY no_o_id ASC

      IF (@@rowcount <> 0)
      BEGIN
         --  claim the order for this district
         DELETE  new_order
         WHERE   no_w_id = @w_id AND
                 no_d_id = @d_id AND
                 no_o_id = @o_id

         --  set carrier_id on this order (and get customer id)
         UPDATE  orders
         SET     o_carrier_id    = @o_carrier_id,
                 @c_id           = o_c_id
         WHERE   o_w_id          = @w_id AND
                 o_d_id          = @d_id AND
                 o_id            = @o_id

         --  set date in all lineitems for this order (and sum amounts)
         UPDATE  order_line
         SET     ol_delivery_d   = GETDATE(),
                 @total          = @total + ol_amount
         WHERE   ol_w_id         = @w_id AND
                 ol_d_id         = @d_id AND
                 ol_o_id         = @o_id

         --  accummulate lineitem amounts for this order into customer
         UPDATE  customer
         SET     c_balance       = c_balance + @total,
                 c_delivery_cnt  = c_delivery_cnt + 1
         WHERE   c_d_id          = @d_id AND
                 c_w_id          = @w_id AND
                 c_id            = @c_id
      END

   SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1
END,
         @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
         @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
         @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
         @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
         @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
         @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
         @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
         @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
         @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
   END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT  @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
GO

-- End SP31


SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
```

createpaymentproc.sql

```
----------------------------------------------------------------------
--                                                                  --
--  File:   PAYMENT.SQL                                             --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                 --
--          Copyright Microsoft, 2005                               --
--          Creates payment stored procedure                        --
--                                                                  --
--          Interface Level:    4.20.000                            --
--                                                                  --
----------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_payment' )
   DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE    tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
```

```
    @c_balance      money,
    @c_discount     smallmoney,
    @c_data         char(42),
    @datetime       datetime,
    @w_ytd          money,
    @d_ytd          money,
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   --  get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   --  get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_zip      = w_zip,
           @w_state    = w_state,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   --  create history record
           INSERT INTO                 history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   --  return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- Create SP0
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp0tpcc_payment' )
    DROP PROCEDURE sp0tpcc_payment
GO
```

```
CREATE PROC sp0tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
   -- get customer id and info using last name
   SELECT  @cnt    = COUNT(*)
   FROM    customer WITH (repeatableread)
   WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


   SELECT  @val    = (@cnt + 1) / 2

   SET     rowcount @val

   SELECT  @c_id   = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
   ORDER   BY c_last, c_first

   SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --       compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   -- update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
```

```
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP0


-- Create SP1
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp1tpcc_payment' )
   DROP PROCEDURE sp1tpcc_payment
GO

CREATE PROC sp1tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND   c_w_id  = @c_w_id AND
                                      c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
```

```
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP1



-- Create SP2
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp2tpcc_payment' )
    DROP PROCEDURE sp2tpcc_payment
GO

CREATE PROC sp2tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id



        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    --  get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
```

```
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
    WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

    -- if customer has bad credit get some more info
    IF (@c_credit = "BC")
    BEGIN
       --       compute new info
       SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

       --  update customer info
       UPDATE  customer
       SET     c_data          = @c_data + substring(c_data, 1, 458),
               @screen_data    = @c_data + substring(c_data, 1, 158)

       WHERE   c_id            = @c_id AND
               c_w_id          = @c_w_id AND
               c_d_id          = @c_d_id
    END

    -- get district data and update year-to-date
    UPDATE  district
    SET     d_ytd       = d_ytd + @h_amount,
            @d_street_1 = d_street_1,
            @d_street_2 = d_street_2,
            @d_city     = d_city,
            @d_state    = d_state,
            @d_zip      = d_zip,
            @d_name     = d_name,
            @d_id_local = d_id
    WHERE   d_w_id      = @w_id AND
            d_id        = @d_id

    -- get warehouse data and update year-to-date
    UPDATE  warehouse
    SET     w_ytd       = w_ytd + @h_amount,
            @w_street_1 = w_street_1,
            @w_street_2 = w_street_2,
            @w_city     = w_city,
            @w_state    = w_state,
            @w_zip      = w_zip,
            @w_name     = w_name,
            @w_id_local = w_id
    WHERE   w_id        = @w_id

    -- create history record
            INSERT INTO             history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    -- return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP2
```

```
-- Create SP3
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp3tpcc_payment' )
    DROP PROCEDURE sp3tpcc_payment
GO

CREATE PROC sp3tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int
```

```
SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   --  get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   --  get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   --  create history record
               INSERT INTO                 history VALUES (@c_id_local,
                           @c_d_id,
                           @c_w_id,
                           @d_id_local,
                           @w_id_local,
                           @datetime,
                           @w_name + '    ' + @d_name)
                           @h_amount,

   COMMIT TRANSACTION p

   --  return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- End SP3



-- Create SP4
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp4tpcc_payment' )
    DROP PROCEDURE sp4tpcc_payment
GO

CREATE PROC sp4tpcc_payment
                   @w_id       int,
                   @c_w_id     int,
                   @h_amount   smallmoney,
                   @d_id       tinyint,
                   @c_d_id     tinyint,
                   @c_id       int,
                   @c_last     char(16) = ""
```

```
AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id

      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)
      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   -- create history record
           INSERT INTO             history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
```

```
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP4



-- Create SP5
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp5tpcc_payment' )
   DROP PROCEDURE sp5tpcc_payment
GO

CREATE PROC sp5tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
      @w_street_2     char(20),
      @w_city         char(20),
      @w_state        char(2),
      @w_zip          char(9),
      @w_name         char(10),
      @d_street_1     char(20),
      @d_street_2     char(20),
      @d_city         char(20),
      @d_state        char(2),
      @d_zip          char(9),
      @d_name         char(10),
      @c_first        char(16),
      @c_middle       char(2),
      @c_street_1     char(20),
      @c_street_2     char(20),
      @c_city         char(20),
      @c_state        char(2),
      @c_zip          char(9),
      @c_phone        char(16),
      @c_since        datetime,
      @c_credit       char(2),
      @c_credit_lim   money,
      @c_balance      money,
      @c_discount     smallmoney,
      @c_data         char(42),
      @datetime       datetime,
      @w_ytd          money,
      @d_ytd          money,
      @cnt            smallint,
      @val            smallint,
      @screen_data    char(200),
      @d_id_local     tinyint,
      @w_id_local     int,
      @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id



      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --        compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                   convert(char(4),@c_d_id) +
                   convert(char(5),@c_w_id) +
                   convert(char(4),@d_id) +
                   convert(char(5),@w_id) +
                   convert(char(19),@h_amount)

   -- update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
```

```
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_discount,
        @c_credit_lim,
        @c_balance,
        @screen_data
GO

-- End SP5



-- Create SP6
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp6tpcc_payment' )
    DROP PROCEDURE sp6tpcc_payment
GO

CREATE PROC sp6tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @d_ytd          money,
        @w_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
```

```
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --       compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   -- update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id
-- create history record
                INSERT INTO             history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   -- return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- End SP6



-- Create SP7
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp7tpcc_payment' )
   DROP PROCEDURE sp7tpcc_payment
GO

CREATE PROC sp7tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
```

```
BEGIN
   --  get customer id and info using last name
   SELECT  @cnt    = COUNT(*)
   FROM    customer WITH (repeatableread)
   WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id


   SELECT  @val    = (@cnt + 1) / 2

   SET     rowcount @val

   SELECT  @c_id   = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id
   ORDER   BY c_last, c_first

   SET     rowcount 0
END

--  get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

--  if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --       compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                   convert(char(4),@c_d_id) +
                   convert(char(4),@c_w_id) +
                   convert(char(5),@w_id) +
                   convert(char(5),@w_id) +
                   convert(char(19),@h_amount)


   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP7



-- Create SP8
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp8tpcc_payment' )
    DROP PROCEDURE sp8tpcc_payment
GO

CREATE PROC sp8tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
```

```
    @w_name         char(10),
    @d_street_1     char(20),
    @d_street_2     char(20),
    @d_city         char(20),
    @d_state        char(2),
    @d_zip          char(9),
    @d_name         char(10),
    @c_first        char(16),
    @c_middle       char(2),
    @c_street_1     char(20),
    @c_street_2     char(20),
    @c_city         char(20),
    @c_state        char(2),
    @c_zip          char(9),
    @c_phone        char(16),
    @c_since        datetime,
    @c_credit       char(2),
    @c_credit_lim   money,
    @c_balance      money,
    @c_discount     smallmoney,
    @c_data         char(42),
    @datetime       datetime,
    @w_ytd          money,
    @d_ytd          money,
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   --  get district data and update year-to-date
   UPDATE  district
   SET     d_ytd           = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   --  get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   --  create history record
           INSERT INTO             history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   --  return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
```

```
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP8



-- Create SP9
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp9tpcc_payment' )
   DROP PROCEDURE sp9tpcc_payment
GO

CREATE PROC sp9tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
      @w_street_2     char(20),
      @w_city         char(20),
      @w_state        char(2),
      @w_zip          char(9),
      @w_name         char(10),
      @d_street_1     char(20),
      @d_street_2     char(20),
      @d_city         char(20),
      @d_state        char(2),
      @d_zip          char(9),
      @d_name         char(10),
      @c_first        char(16),
      @c_middle       char(2),
      @c_street_1     char(20),
      @c_street_2     char(20),
      @c_city         char(20),
      @c_state        char(2),
      @c_zip          char(9),
      @c_since        datetime,
      @c_phone        char(16),
      @c_credit       char(2),
      @c_credit_lim   money,
      @c_balance      money,
      @c_discount     smallmoney,
      @c_data         char(42),
      @datetime       datetime,
      @w_ytd          money,
      @d_ytd          money,
      @cnt            smallint,
      @val            smallint,
      @screen_data    char(200),
      @d_id_local     tinyint,
      @w_id_local     int,
      @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
         c_payment_cnt   = c_payment_cnt + 1,
         c_ytd_payment   = c_ytd_payment + @h_amount,
         @c_first        = c_first,
         @c_middle       = c_middle,
         @c_street_1     = c_street_1,
         @c_street_2     = c_street_2,
         @c_city         = c_city,
         @c_state        = c_state,
         @c_zip          = c_zip,
         @c_phone        = c_phone,
         @c_credit       = c_credit,
         @c_credit_lim   = c_credit_lim,
         @c_discount     = c_discount,
         @c_since        = c_since,
         @c_id_local     = c_id
   WHERE   c_id          = @c_id AND
         c_w_id          = @c_w_id AND
         c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id          = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
         @d_street_1 = d_street_1,
         @d_street_2 = d_street_2,
         @d_city     = d_city,
         @d_state    = d_state,
         @d_zip      = d_zip,
         @d_name     = d_name,
         @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
         d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
         @w_street_1 = w_street_1,
```

```
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
    WHERE   w_id        = @w_id

    --  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    --  return data to client
    SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP9
```

```
-- Create SP10
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp10tpcc_payment' )
    DROP PROCEDURE sp10tpcc_payment
GO

CREATE PROC sp10tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int
```

```
SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    --  get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
            @c_street_1     = c_street_1,
            @c_street_2     = c_street_2,
            @c_city         = c_city,
            @c_state        = c_state,
            @c_zip          = c_zip,
            @c_phone        = c_phone,
            @c_credit       = c_credit,
            @c_credit_lim   = c_credit_lim,
            @c_discount     = c_discount,
            @c_since        = c_since,
            @c_id_local     = c_id
    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
```

```
-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --       compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @h_amount,
                        @datetime,
                        @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   -- return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- End SP10


-- Create SP11
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp11tpcc_payment' )
   DROP PROCEDURE sp11tpcc_payment
GO

CREATE PROC sp11tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""
AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                  c_w_id  = @c_w_id AND
                                  c_d_id  = @c_d_id
```

```
   SELECT  @val    = (@cnt + 1) / 2

   SET     rowcount @val

   SELECT  @c_id   = c_id
   FROM    customer WITH (repeatableread)
   WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id
   ORDER   BY c_last, c_first

   SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --        compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP11



-- Create SP12
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp12tpcc_payment' )
    DROP PROCEDURE sp12tpcc_payment
GO

CREATE PROC sp12tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
```

```
    @c_middle       char(2),
    @c_street_1     char(20),
    @c_street_2     char(20),
    @c_city         char(20),
    @c_state        char(2),
    @c_zip          char(9),
    @c_phone        char(16),
    @c_since        datetime,
    @c_credit       char(2),
    @c_credit_lim   money,
    @c_balance      money,
    @c_discount     smallmoney,
    @c_data         char(42),
    @datetime       datetime,
    @w_ytd          money,
    @d_ytd          money,
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
  -- get payment date
  SELECT  @datetime = GETDATE()

  IF (@c_id = 0)
  BEGIN
    -- get customer id and info using last name
    SELECT  @cnt    = COUNT(*)
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id


    SELECT  @val    = (@cnt + 1) / 2

    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id
    ORDER   BY c_last, c_first

    SET     rowcount 0
  END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --          compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_w_id_local = d_w_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
        INSERT INTO             history VALUES (@c_id_local,
                @c_d_id,
                @c_w_id,
                @d_id_local,
                @w_id_local,
                @datetime,
                @h_amount,
                @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
```

```
        @c_balance,
        @screen_data
GO

-- End SP12



-- Create SP13
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp13tpcc_payment' )
   DROP PROCEDURE sp13tpcc_payment
GO

CREATE PROC sp13tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @d_ytd          money,
        @w_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id
```

```
    -- create history record
            INSERT INTO             history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    -- return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP13



-- Create SP14
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp14tpcc_payment' )
    DROP PROCEDURE sp14tpcc_payment
GO
```

```
CREATE PROC sp14tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    -- get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
```

```
BEGIN -- get customer id and info using last name
    SELECT  @cnt    = COUNT(*)
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id


    SELECT  @val    = (@cnt + 1) / 2

    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id
    ORDER   BY c_last, c_first

    SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --      compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                    convert(char(4),@c_d_id) +
                    convert(char(5),@c_w_id) +
                    convert(char(4),@d_id) +
```

```
                convert(char(5),@w_id) +
                convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    --  return data to client
    SELECT  @c_id,
            @c_last,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP14



-- Create SP15
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp15tpcc_payment' )
    DROP PROCEDURE sp15tpcc_payment
GO

CREATE PROC sp15tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
```

```
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id
   ORDER   BY c_last, c_first

   SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
   --        compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

-- get district data and update year-to-date
SET     d_ytd
UPDATE  district  = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_state,
        @c_city,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP15



-- Create SP16
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp16tpcc_payment' )
   DROP PROCEDURE sp16tpcc_payment
GO

CREATE PROC sp16tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
```

```
    @c_credit       char(2),
    @c_credit_lim   money,
    @c_balance      money,
    @c_discount     smallmoney,
    @c_data         char(42),
    @datetime       datetime,
    @w_ytd          money,
    @d_ytd          money,
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   -- create history record
           INSERT INTO             history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   -- return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- End SP16
```

```
-- Create SP17
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp17tpcc_payment' )
    DROP PROCEDURE sp17tpcc_payment
GO

CREATE PROC sp17tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    -- get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        -- get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id


        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    -- get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
            @c_street_1     = c_street_1,
            @c_street_2     = c_street_2,
            @c_city         = c_city,
            @c_state        = c_state,
            @c_zip          = c_zip,
            @c_phone        = c_phone,
            @c_credit       = c_credit,
            @c_credit_lim   = c_credit_lim,
            @c_discount     = c_discount,
            @c_since        = c_since,
            @c_id_local     = c_id
    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id

    IF (@c_credit = "BC")
    BEGIN
        --      compute new info
        SELECT  @c_data =   convert(char(5),@c_id) +
                            convert(char(4),@c_d_id) +
                            convert(char(5),@c_w_id) +
                            convert(char(4),@d_id) +
                            convert(char(5),@w_id) +
                            convert(char(19),@h_amount)

        -- update customer info
        UPDATE  customer
        SET     c_data          = @c_data + substring(c_data, 1, 458),
                @screen_data    = @c_data + substring(c_data, 1, 158)

        WHERE   c_id            = @c_id AND
                c_w_id          = @c_w_id AND
                c_d_id          = @c_d_id
    END

    -- get district data and update year-to-date
    UPDATE  district
    SET     d_ytd       = d_ytd + @h_amount,
            @d_street_1 = d_street_1,
            @d_street_2 = d_street_2,
            @d_city     = d_city,
            @d_state    = d_state,
            @d_zip      = d_zip,
            @d_name     = d_name,
            @d_id_local = d_id
    WHERE   d_w_id      = @w_id AND
            d_id        = @d_id

    -- get warehouse data and update year-to-date
    UPDATE  warehouse
    SET     w_ytd       = w_ytd + @h_amount,
            @w_street_1 = w_street_1,
            @w_street_2 = w_street_2,
            @w_city     = w_city,
            @w_state    = w_state,
            @w_zip      = w_zip,
            @w_name     = w_name,
            @w_id_local = w_id
    WHERE   w_id        = @w_id

    -- create history record
            INSERT INTO             history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
```

```
                    @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    -- return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP17



-- Create SP18
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp18tpcc_payment' )
    DROP PROCEDURE sp18tpcc_payment
GO

CREATE PROC sp18tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""
AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    -- get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        -- get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id

        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    -- get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
            @c_street_1     = c_street_1,
            @c_street_2     = c_street_2,
            @c_city         = c_city,
            @c_state        = c_state,
            @c_zip          = c_zip,
            @c_phone        = c_phone,
            @c_credit       = c_credit,
            @c_credit_lim   = c_credit_lim,
            @c_discount     = c_discount,
            @c_since        = c_since,
            @c_id_local     = c_id
    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id

    -- if customer has bad credit get some more info
    IF (@c_credit = "BC")
    BEGIN
        --      compute new info
        SELECT  @c_data =   convert(char(5),@c_id) +
                            convert(char(4),@c_d_id) +
                            convert(char(5),@c_w_id) +
                            convert(char(4),@d_id) +
                            convert(char(5),@w_id) +
                            convert(char(19),@h_amount)

        -- update customer info
        UPDATE  customer
        SET     c_data          = @c_data + substring(c_data, 1, 458),
                @screen_data    = @c_data + substring(c_data, 1, 158)
```

```
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_city,
        @d_street_2,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP18



-- Create SP19
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp19tpcc_payment' )
    DROP PROCEDURE sp19tpcc_payment
GO

CREATE PROC sp19tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
```

```
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

--  if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --        compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
WHERE   d_w_id      = @w_id AND
        @d_id_local = d_id
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
                @screen_data
                @c_balance)
GO

-- End SP19



-- Create SP20
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp20tpcc_payment' )
    DROP PROCEDURE sp20tpcc_payment
GO

CREATE PROC sp20tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
```

```
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
  -- get payment date
  SELECT  @datetime = GETDATE()

  IF (@c_id = 0)
  BEGIN
    -- get customer id and info using last name
    SELECT  @cnt    = COUNT(*)
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


    SELECT  @val    = (@cnt + 1) / 2

    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
    ORDER   BY c_last, c_first

    SET     rowcount 0
  END

  -- get customer info and update balances
  UPDATE  customer
  SET     @c_balance      = c_balance   = c_balance - @h_amount,
          c_payment_cnt   = c_payment_cnt + 1,
          c_ytd_payment   = c_ytd_payment + @h_amount,
          @c_first        = c_first,
          @c_middle       = c_middle,
          @c_last         = c_last,
          @c_street_1     = c_street_1,
          @c_street_2     = c_street_2,
          @c_city         = c_city,
          @c_state        = c_state,
          @c_zip          = c_zip,
          @c_phone        = c_phone,
          @c_credit       = c_credit,
          @c_discount     = c_discount,
          @c_credit_lim   = c_credit_lim,
          @c_since        = c_since,
          @c_id_local     = c_id
  WHERE   c_id            = @c_id AND
          c_w_id          = @c_w_id AND
          c_d_id          = @c_d_id

  -- if customer has bad credit get some more info
  IF (@c_credit = "BC")
  BEGIN
    --        compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
  END

  -- get district data and update year-to-date
  UPDATE  district
  SET     d_ytd       = d_ytd + @h_amount,
          @d_street_1 = d_street_1,
          @d_street_2 = d_street_2,
          @d_city     = d_city,
          @d_state    = d_state,
          @d_zip      = d_zip,
          @d_name     = d_name,
          @d_id_local = d_id
  WHERE   d_w_id      = @w_id AND
          d_id        = @d_id

  -- get warehouse data and update year-to-date
  UPDATE  warehouse
  SET     w_ytd       = w_ytd + @h_amount,
          @w_street_1 = w_street_1,
          @w_street_2 = w_street_2,
          @w_city     = w_city,
          @w_state    = w_state,
          @w_zip      = w_zip,
          @w_name     = w_name,
          @w_id_local = w_id
  WHERE   w_id        = @w_id

  -- create history record
  INSERT INTO             history VALUES (@c_id_local,
                          @c_d_id,
                          @c_w_id,
                          @d_id_local,
                          @w_id_local,
                          @datetime,
                          @h_amount,
                          @w_name + '    ' + @d_name)

  COMMIT TRANSACTION p

  -- return data to client
  SELECT  @c_id,
          @c_last,
          @datetime,
          @w_street_1,
          @w_street_2,
          @w_city,
          @w_state,
          @w_zip,
          @d_street_1,
          @d_street_2,
          @d_city,
          @d_state,
          @d_zip,
          @c_first,
          @c_middle,
          @c_street_1,
          @c_street_2,
          @c_city,
          @c_state,
          @c_zip,
          @c_phone,
          @c_since,
          @c_credit,
          @c_credit_lim,
          @c_discount,
          @c_balance,
          @screen_data
GO

-- End SP20

-- Create SP21
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp21tpcc_payment' )
    DROP PROCEDURE sp21tpcc_payment
GO

CREATE PROC sp21tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
```

```
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt = count(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)
      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   -- create history record
           INSERT INTO                 history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
```

```
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP21



-- Create SP22
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp22tpcc_payment' )
   DROP PROCEDURE sp22tpcc_payment
GO

CREATE PROC sp22tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
      @w_street_2     char(20),
      @w_city         char(20),
      @w_state        char(2),
      @w_zip          char(9),
      @w_name         char(10),
      @d_street_1     char(20),
      @d_street_2     char(20),
      @d_city         char(20),
      @d_state        char(2),
      @d_zip          char(9),
      @d_name         char(10),
      @c_first        char(16),
      @c_middle       char(2),
      @c_street_1     char(20),
      @c_street_2     char(20),
      @c_city         char(20),
      @c_state        char(2),
      @c_zip          char(9),
      @c_phone        char(16),
      @c_since        datetime,
      @c_credit       char(2),
      @c_credit_lim   money,
      @c_balance      money,
      @c_discount     smallmoney,
      @c_data         char(42),
      @datetime       datetime,
      @w_ytd          money,
      @d_ytd          money,
      @cnt            smallint,
      @val            smallint,
      @screen_data    char(200),
      @d_id_local     tinyint,
      @w_id_local     int,
      @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id



      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first
      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
         c_payment_cnt   = c_payment_cnt + 1,
         c_ytd_payment   = c_ytd_payment + @h_amount,
         @c_first        = c_first,
         @c_middle       = c_middle,
         @c_last         = c_last,
         @c_street_1     = c_street_1,
         @c_street_2     = c_street_2,
         @c_city         = c_city,
         @c_state        = c_state,
         @c_zip          = c_zip,
         @c_phone        = c_phone,
         @c_credit       = c_credit,
         @c_credit_lim   = c_credit_lim,
         @c_discount     = c_discount,
         @c_since        = c_since,
         @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
         c_w_id          = @c_w_id AND
         c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
         @d_street_1 = d_street_1,
         @d_street_2 = d_street_2,
```

```
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_phone,
        @c_zip,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP22



-- Create SP23
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp23tpcc_payment' )
    DROP PROCEDURE sp23tpcc_payment
GO

CREATE PROC sp23tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_discount     smallmoney,
        @c_balance      money,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id


        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                            c_w_id  = @c_w_id AND
                                            c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    --  get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
            @c_street_1     = c_street_1,
            @c_street_2     = c_street_2,
            @c_city         = c_city,
```

```
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --      compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_zip      = w_zip,
        @w_state    = w_state,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
        INSERT INTO             history VALUES (@c_id_local,
                @c_d_id,
                @c_w_id,
                @d_id_local,
                @w_id_local,
                @datetime,
                @h_amount,
                @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    -- return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP23



-- Create SP24

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp24tpcc_payment' )  DROP PROCEDURE sp24tpcc_payment
GO

CREATE PROC sp24tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""
```

```
BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
```

```
BEGIN
IF (@c_credit = "BC")
   --         compute new info
   SELECT  @c_data =   convert(char(5),@c_id) +
                       convert(char(4),@c_d_id) +
                       convert(char(5),@c_w_id) +
                       convert(char(4),@d_id) +
                       convert(char(5),@w_id) +
                       convert(char(19),@h_amount)

   --  update customer info
   UPDATE  customer
   SET     c_data          = @c_data + substring(c_data, 1, 458),
           @screen_data    = @c_data + substring(c_data, 1, 158)

   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                    @c_d_id,
                    @c_w_id,
                    @d_id_local,
                    @w_id_local,
                    @datetime,
                    @h_amount,
                    @w_name + '    ' + @d_name)
```

```
   COMMIT TRANSACTION p

   --  return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
           @d_zip,
           @c_first,
           @c_middle,
           @c_street_1,
           @c_street_2,
           @c_city,
           @c_state,
           @c_zip,
           @c_phone,
           @c_since,
           @c_credit,
           @c_credit_lim,
           @c_discount,
           @c_balance,
           @screen_data
GO

-- End SP24



-- Create SP25
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp25tpcc_payment' )
   DROP PROCEDURE sp25tpcc_payment
GO

CREATE PROC sp25tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
```

```
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   --  get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      --  get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id

      SELECT  @val    = (@cnt + 1) / 2
      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   --  get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   --  if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      --  update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_d_id          = @c_d_id AND
              c_w_id          = @c_w_id
   END

   --  get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   --  get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
           @w_zip      = w_zip,
           @w_name     = w_name,
           @w_id_local = w_id
   WHERE   w_id        = @w_id

   --  create history record
           INSERT INTO             history VALUES (@c_id_local,
                   @c_d_id,
                   @c_w_id,
                   @d_id_local,
                   @w_id_local,
                   @datetime,
                   @h_amount,
                   @w_name + '    ' + @d_name)

   COMMIT TRANSACTION p

   --  return data to client
   SELECT  @c_id,
           @c_last,
           @datetime,
           @w_street_1,
           @w_street_2,
           @w_city,
           @w_state,
           @w_zip,
           @d_street_1,
           @d_street_2,
           @d_city,
           @d_state,
```

```
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP25



-- Create SP26
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp26tpcc_payment' )
   DROP PROCEDURE sp26tpcc_payment
GO

CREATE PROC sp26tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
      @w_street_2     char(20),
      @w_city         char(20),
      @w_state        char(2),
      @w_zip          char(9),
      @w_name         char(10),
      @d_street_1     char(20),
      @d_street_2     char(20),
      @d_city         char(20),
      @d_state        char(2),
      @d_zip          char(9),
      @d_name         char(10),
      @c_first        char(16),
      @c_middle       char(2),
      @c_street_1     char(20),
      @c_street_2     char(20),
      @c_city         char(20),
      @c_state        char(2),
      @c_zip          char(9),
      @c_phone        char(16),
      @c_since        datetime,
      @c_credit       char(2),
      @c_credit_lim   money,
      @c_balance      money,
      @c_discount     smallmoney,
      @c_data         char(42),
      @datetime       datetime,
      @w_ytd          money,
      @d_ytd          money,
      @cnt            smallint,
      @val            smallint,
      @screen_data    char(200),
      @d_id_local     tinyint,
      @w_id_local     int,
      @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     c_payment_cnt   = c_payment_cnt + 1,
           c_balance       = c_balance - @h_amount,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_city         = c_city,
           @c_state        = c_state,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id
```

```
-- get warehouse data and update year-to-date
UPDATE warehouse
SET     w_ytd       = w_ytd + @h_amount,
       @w_street_1 = w_street_1,
       @w_street_2 = w_street_2,
       @w_city     = w_city,
       @w_state    = w_state,
       @w_zip      = w_zip,
       @w_name     = w_name,
       @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,
       @c_discount,
       @c_balance,
GO      @screen_data

-- End SP26



-- Create SP27
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp27tpcc_payment' )
   DROP PROCEDURE sp27tpcc_payment
GO

CREATE PROC sp27tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
       @w_street_2     char(20),
       @w_city         char(20),
       @w_state        char(2),
       @w_zip          char(9),
       @w_name         char(10),
       @d_street_1     char(20),
       @d_street_2     char(20),
       @d_city         char(20),
       @d_state        char(2),
       @d_zip          char(9),
       @d_name         char(10),
       @c_first        char(16),
       @c_middle       char(2),
       @c_street_1     char(20),
       @c_street_2     char(20),
       @c_city         char(20),
       @c_state        char(2),
       @c_zip          char(9),
       @c_phone        char(16),
       @c_since        datetime,
       @c_credit       char(2),
       @c_credit_lim   money,
       @c_balance      money,
       @c_discount     smallmoney,
       @c_data         char(42),
       @datetime       datetime,
       @w_ytd          money,
       @d_ytd          money,
       @cnt            smallint,
       @screen_data    char(200),
       @err            smallint,
       @d_id_local     tinyint,
       @w_id_local     int,
       @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
          c_payment_cnt   = c_payment_cnt + 1,
          c_ytd_payment   = c_ytd_payment + @h_amount,
          @c_first        = c_first,
          @c_middle       = c_middle,
          @c_last         = c_last,
          @c_street_1     = c_street_1,
          @c_street_2     = c_street_2,
          @c_city         = c_city,
          @c_state        = c_state,
          @c_zip          = c_zip,
          @c_phone        = c_phone,
          @c_credit       = c_credit,
          @c_credit_lim   = c_credit_lim,
          @c_discount     = c_discount,
          @c_since        = c_since,
```

```
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

--  if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --          compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record

                INSERT INTO history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    --  return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

-- End SP27



-- Create SP28
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp28tpcc_payment' )
    DROP PROCEDURE sp28tpcc_payment
GO

CREATE PROC sp28tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @d_id       tinyint,
                    @h_amount   smallmoney,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
```

```
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


    SELECT  @val    = (@cnt + 1) / 2

    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
    ORDER   BY c_last, c_first

    SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --      compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)
    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP28



-- Create SP29
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp29tpcc_payment' )
    DROP PROCEDURE sp29tpcc_payment
GO

CREATE PROC sp29tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
```

```
    @d_city         char(20),
    @d_state        char(2),
    @d_zip          char(9),
    @d_name         char(10),
    @c_first        char(16),
    @c_middle       char(2),
    @c_street_1     char(20),
    @c_street_2     char(20),
    @c_city         char(20),
    @c_state        char(2),
    @c_zip          char(9),
    @c_phone        char(16),
    @c_since        datetime,
    @c_credit       char(2),
    @c_credit_lim   money,
    @c_balance      money,
    @c_discount     smallmoney,
    @c_data         char(42),
    @datetime       datetime,
    @w_ytd          money,
    @d_ytd          money,
    @cnt            smallint,
    @val            smallint,
    @screen_data    char(200),
    @d_id_local     tinyint,
    @w_id_local     int,
    @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
  -- get payment date
  SELECT  @datetime = GETDATE()

  IF (@c_id = 0)
  BEGIN
    -- get customer id and info using last name
    SELECT  @cnt    = COUNT(*)
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
                                    c_d_id  = @c_d_id


    SELECT  @val    = (@cnt + 1) / 2

    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                                    c_w_id  = @c_w_id AND
  ORDER   BY c_last, c_first        c_d_id  = @c_d_id

  SET     rowcount 0
END

-- get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance   = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --      compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)

    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

-- get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_city     = d_city,
        @d_street_2 = d_street_2,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-- get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

-- create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
```

```
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP29



-- Create SP30
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp30tpcc_payment' )
   DROP PROCEDURE sp30tpcc_payment
GO

CREATE PROC sp30tpcc_payment
                @w_id       int,
                @c_w_id     int,
                @h_amount   smallmoney,
                @d_id       tinyint,
                @c_d_id     tinyint,
                @c_id       int,
                @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_balance      money,
        @c_credit_lim   money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
   -- get payment date
   SELECT  @datetime = GETDATE()

   IF (@c_id = 0)
   BEGIN
      -- get customer id and info using last name
      SELECT  @cnt    = COUNT(*)
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id


      SELECT  @val    = (@cnt + 1) / 2

      SET     rowcount @val

      SELECT  @c_id   = c_id
      FROM    customer WITH (repeatableread)
      WHERE   c_last  = @c_last AND
                                          c_w_id  = @c_w_id AND
                                          c_d_id  = @c_d_id
      ORDER   BY c_last, c_first

      SET     rowcount 0
   END

   -- get customer info and update balances
   UPDATE  customer
   SET     @c_balance      = c_balance   = c_balance - @h_amount,
           c_payment_cnt   = c_payment_cnt + 1,
           c_ytd_payment   = c_ytd_payment + @h_amount,
           @c_first        = c_first,
           @c_middle       = c_middle,
           @c_last         = c_last,
           @c_street_1     = c_street_1,
           @c_street_2     = c_street_2,
           @c_state        = c_state,
           @c_city         = c_city,
           @c_zip          = c_zip,
           @c_phone        = c_phone,
           @c_credit       = c_credit,
           @c_credit_lim   = c_credit_lim,
           @c_discount     = c_discount,
           @c_since        = c_since,
           @c_id_local     = c_id
   WHERE   c_id            = @c_id AND
           c_w_id          = @c_w_id AND
           c_d_id          = @c_d_id

   -- if customer has bad credit get some more info
   IF (@c_credit = "BC")
   BEGIN
      --        compute new info
      SELECT  @c_data =   convert(char(5),@c_id) +
                          convert(char(4),@c_d_id) +
                          convert(char(5),@c_w_id) +
                          convert(char(4),@d_id) +
                          convert(char(5),@w_id) +
                          convert(char(19),@h_amount)

      -- update customer info
      UPDATE  customer
      SET     c_data          = @c_data + substring(c_data, 1, 458),
              @screen_data    = @c_data + substring(c_data, 1, 158)

      WHERE   c_id            = @c_id AND
              c_w_id          = @c_w_id AND
              c_d_id          = @c_d_id
   END

   -- get district data and update year-to-date
   UPDATE  district
   SET     d_ytd       = d_ytd + @h_amount,
           @d_street_1 = d_street_1,
           @d_street_2 = d_street_2,
           @d_city     = d_city,
           @d_state    = d_state,
           @d_zip      = d_zip,
           @d_name     = d_name,
           @d_id_local = d_id
   WHERE   d_w_id      = @w_id AND
           d_id        = @d_id

   -- get warehouse data and update year-to-date
   UPDATE  warehouse
   SET     w_ytd       = w_ytd + @h_amount,
           @w_street_1 = w_street_1,
           @w_street_2 = w_street_2,
           @w_city     = w_city,
           @w_state    = w_state,
```

```
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
    WHERE   w_id        = @w_id

    --  create history record
                INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

    COMMIT TRANSACTION p

    --  return data to client
    SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP30


-- Create SP31

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp31tpcc_payment' )  DROP PROCEDURE sp31tpcc_payment
GO

CREATE PROC sp31tpcc_payment
                    @w_id       int,
                    @c_w_id     int,
                    @h_amount   smallmoney,
                    @d_id       tinyint,
                    @c_d_id     tinyint,
                    @c_id       int,
                    @c_last     char(16) = ""


AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id


        SELECT  @val    = (@cnt + 1) / 2

        SET     rowcount @val

        SELECT  @c_id   = c_id
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                                        c_w_id  = @c_w_id AND
                                        c_d_id  = @c_d_id
        ORDER   BY c_last, c_first

        SET     rowcount 0
    END

    --  get customer info and update balances
    UPDATE  customer
    SET     @c_balance      = c_balance   = c_balance - @h_amount,
            c_payment_cnt   = c_payment_cnt + 1,
            c_ytd_payment   = c_ytd_payment + @h_amount,
            @c_first        = c_first,
            @c_middle       = c_middle,
            @c_last         = c_last,
            @c_street_1     = c_street_1,
            @c_street_2     = c_street_2,
            @c_city         = c_city,
            @c_state        = c_state,
            @c_zip          = c_zip,
            @c_phone        = c_phone,
            @c_credit       = c_credit,
            @c_credit_lim   = c_credit_lim,
            @c_discount     = c_discount,
            @c_since        = c_since,
            @c_id_local     = c_id
    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id

    --  if customer has bad credit get some more info
    IF (@c_credit = "BC")
    BEGIN
```

```
--        compute new info
SELECT  @c_data =   convert(char(5),@c_id) +
                    convert(char(4),@c_d_id) +
                    convert(char(5),@c_w_id) +
                    convert(char(4),@d_id) +
                    convert(char(5),@w_id) +
                    convert(char(19),@h_amount)

--  update customer info
UPDATE  customer
SET     c_data          = @c_data + substring(c_data, 1, 458),
        @screen_data    = @c_data + substring(c_data, 1, 158)

WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state    = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
            INSERT INTO                 history VALUES (@c_id_local,
                        @c_d_id,
                        @c_w_id,
                        @d_id_local,
                        @w_id_local,
                        @datetime,
                        @h_amount,
                        @w_name + '    ' + @d_name)

COMMIT TRANSACTION p

--  return data to client
SELECT  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

-- End SP31


SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
```

createstocklevproc.sql

```
--------------------------------------------------------------------
--                                                                --
--  File:   STOCKLEV.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.62     --      --
--          Copyright Microsoft, 2005                       --
--                                                          --
--          Creates stock level stored procedure            --
--                                                          --
--          Interface Level:    4.20.000                    --
--                                                          --
--------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_stocklevel'
)
    DROP PROCEDURE tpcc_stocklevel
GO

CREATE PROCEDURE    tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO

-- Create SP0
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp0tpcc_stocklevel' )
```

```
   DROP PROCEDURE sp0tpcc_stocklevel
GO

CREATE PROC sp0tpcc_stocklevel
                @w_id           int,
                @d_id           tinyint,
                @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                    @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP0


-- Create SP1
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp1tpcc_stocklevel' )
   DROP PROCEDURE sp1tpcc_stocklevel
GO

CREATE PROC sp1tpcc_stocklevel
                @w_id           int,
                @d_id           tinyint,
                @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                    @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP1


-- Create SP2
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp2tpcc_stocklevel' )
   DROP PROCEDURE sp2tpcc_stocklevel
GO

CREATE PROC sp2tpcc_stocklevel
                @w_id           int,
                @d_id           tinyint,
                @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                    @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP2


-- Create SP3
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp3tpcc_stocklevel' )
   DROP PROCEDURE sp3tpcc_stocklevel
GO

CREATE PROC sp3tpcc_stocklevel
                @w_id           int,
                @d_id           tinyint,
                @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                    @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP3


-- Create SP4
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp4tpcc_stocklevel' )
   DROP PROCEDURE sp4tpcc_stocklevel
GO

CREATE PROC sp4tpcc_stocklevel
                @w_id           int,
                @d_id           tinyint,
                @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
```

```
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP4



-- Create SP5
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp5tpcc_stocklevel' )
   DROP PROCEDURE sp5tpcc_stocklevel
GO

CREATE PROC sp5tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP5



-- Create SP6
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp6tpcc_stocklevel' )
   DROP PROCEDURE sp6tpcc_stocklevel
GO

CREATE PROC sp6tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP6



-- Create SP7
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp7tpcc_stocklevel' )
   DROP PROCEDURE sp7tpcc_stocklevel
GO

CREATE PROC sp7tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP7



-- Create SP8
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp8tpcc_stocklevel' )
   DROP PROCEDURE sp8tpcc_stocklevel
GO

CREATE PROC sp8tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
```

```
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP8



-- Create SP9
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp9tpcc_stocklevel' )
    DROP PROCEDURE sp9tpcc_stocklevel
GO

CREATE PROC sp9tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP9



-- Create SP10
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp10tpcc_stocklevel' )
    DROP PROCEDURE sp10tpcc_stocklevel
GO

CREATE PROC sp10tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP10



-- Create SP11
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp11tpcc_stocklevel' )
    DROP PROCEDURE sp11tpcc_stocklevel
GO

CREATE PROC sp11tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP11



-- Create SP12
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp12tpcc_stocklevel' )
    DROP PROCEDURE sp12tpcc_stocklevel
GO

CREATE PROC sp12tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP12



-- Create SP13
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp13tpcc_stocklevel' )
    DROP PROCEDURE sp13tpcc_stocklevel
```

```
GO

CREATE PROC sp13tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP13



-- Create SP14
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp14tpcc_stocklevel' )
    DROP PROCEDURE sp14tpcc_stocklevel
GO

CREATE PROC sp14tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id
SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP14



-- Create SP15
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp15tpcc_stocklevel' )
    DROP PROCEDURE sp15tpcc_stocklevel
GO

CREATE PROC sp15tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP15



-- Create SP16
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp16tpcc_stocklevel' )
    DROP PROCEDURE sp16tpcc_stocklevel
GO

CREATE PROC sp16tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP16



-- Create SP17
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp17tpcc_stocklevel' )
    DROP PROCEDURE sp17tpcc_stocklevel
GO

CREATE PROC sp17tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
```

```
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
      d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
      order_line
WHERE   ol_w_id     = @w_id AND
      ol_d_id     = @d_id and
      ol_o_id     BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id      = ol_w_id AND
      s_i_id      = ol_i_id AND
      s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP17



-- Create SP18
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp18tpcc_stocklevel' )
   DROP PROCEDURE sp18tpcc_stocklevel
GO

CREATE PROC sp18tpcc_stocklevel
                  @w_id           int,
                  @d_id           tinyint,
                  @threshhold     smallint

AS
DECLARE @o_id_low   int,
      @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
      d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
      order_line
WHERE   ol_w_id     = @w_id AND
      ol_d_id     = @d_id and
      ol_o_id     BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id      = ol_w_id AND
      s_i_id      = ol_i_id AND
      s_quantity  < @threshhold
GO
OPTION(ORDER GROUP)
-- End SP18



-- Create SP19
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp19tpcc_stocklevel' )
   DROP PROCEDURE sp19tpcc_stocklevel
GO

CREATE PROC sp19tpcc_stocklevel
                  @w_id           int,
                  @d_id           tinyint,
                  @threshhold     smallint

AS
DECLARE @o_id_low   int,
      @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
      d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
      order_line
WHERE   ol_w_id     = @w_id AND
      ol_d_id     = @d_id and
      ol_o_id     BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id      = ol_w_id AND
      s_i_id      = ol_i_id AND
      s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP19



-- Create SP20
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp20tpcc_stocklevel' )
   DROP PROCEDURE sp20tpcc_stocklevel
GO

CREATE PROC sp20tpcc_stocklevel
                  @w_id           int,
                  @d_id           tinyint,
                  @threshhold     smallint

AS
DECLARE @o_id_low   int,
      @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
      d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
      order_line
WHERE   ol_w_id     = @w_id AND
      ol_d_id     = @d_id and
      ol_o_id     BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id      = ol_w_id AND
      s_i_id      = ol_i_id AND
      s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP20

-- Create SP21
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp21tpcc_stocklevel' )
   DROP PROCEDURE sp21tpcc_stocklevel
GO

CREATE PROC sp21tpcc_stocklevel
                  @w_id           int,
                  @d_id           tinyint,
                  @threshhold     smallint

AS
DECLARE @o_id_low   int,
      @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
      @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
      d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
      order_line
WHERE   ol_w_id     = @w_id AND
      ol_d_id     = @d_id and
      ol_o_id     BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id      = ol_w_id AND
      s_i_id      = ol_i_id AND
```

```
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP21



-- Create SP22
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp22tpcc_stocklevel' )
   DROP PROCEDURE sp22tpcc_stocklevel
GO

CREATE PROC sp22tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP22



-- Create SP23
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp23tpcc_stocklevel' )
   DROP PROCEDURE sp23tpcc_stocklevel
GO

CREATE PROC sp23tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP23



-- Create SP24
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp24tpcc_stocklevel' )
   DROP PROCEDURE sp24tpcc_stocklevel
GO

CREATE PROC sp24tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP24



-- Create SP25
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp25tpcc_stocklevel' )
   DROP PROCEDURE sp25tpcc_stocklevel
GO

CREATE PROC sp25tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP25



-- Create SP26
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp26tpcc_stocklevel' )
   DROP PROCEDURE sp26tpcc_stocklevel
GO
```

```
CREATE PROC sp26tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP26



-- Create SP27
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp27tpcc_stocklevel' )
    DROP PROCEDURE sp27tpcc_stocklevel
GO

CREATE PROC sp27tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP27



-- Create SP28
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp28tpcc_stocklevel' )
    DROP PROCEDURE sp28tpcc_stocklevel
GO

CREATE PROC sp28tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP28
```



```
-- Create SP29
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp29tpcc_stocklevel' )
    DROP PROCEDURE sp29tpcc_stocklevel
GO

CREATE PROC sp29tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP29



-- Create SP30
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp30tpcc_stocklevel' )
    DROP PROCEDURE sp30tpcc_stocklevel
GO

CREATE PROC sp30tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
```

```
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO
-- End SP30


-- Create SP31
IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'sp31tpcc_stocklevel' )
    DROP PROCEDURE sp31tpcc_stocklevel
GO

CREATE PROC sp31tpcc_stocklevel
                    @w_id           int,
                    @d_id           tinyint,
                    @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                        @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO

-- End SP31


SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
```

Appendix C Tunable Parameters

Disabled Windows Services

Alerter
ClipBook
Distributed Link Tracking Client
HID Input Service
IMAPI CD-Burning COM Service
Indexing Service
Intersite Messaging
Kerberos Key Distribution Center
License Logging Service
Messenger
Netmeeting Remote Desktop Sharing
Network DDE
Network DDE DSDM
Routing and Remote Access
Terminal Services Session Directory
WebClient
Windows Audio
Windows Image Acquisition

Server System Configuratiopn

System Information report written at: 11/21/05 18:17:41
System Name: SQLORCA
[System Summary]

Item Value
OS Name Microsoft(R) Windows(R) Server 2003, Datacenter Edition for 64-Bit Itanium-based Systems
Version 5.2.3790 Service Pack 1 Build 3790
Other OS Description Not Available
OS Manufacturer Microsoft Corporation
System Name SQLORCA
System Manufacturer hp
System Model superdome server SD64A
System Type Itanium (TM) -based System
Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

Processor ia64 Family 31 Model 2 Stepping 1 GenuineIntel ~1600 Mhz

BIOS Version/Date HP 2.50, Not Available
SMBIOS Version 2.3
Windows Directory C:\WIN1828
System Directory C:\WIN1828\system32
Boot Device \Device\HarddiskVolume1
Locale United States
Hardware Abstraction Layer Version = "5.2.3790.1828 (srv03_sp1_rtm.050321-2020)"
User Name Not Available
Time Zone Pacific Standard Time
Total Physical Memory 1,048,317.20 MB
Available Physical Memory 1,015.41 GB
Total Virtual Memory 1,016.22 GB
Available Virtual Memory 1,015.33 GB
Page File Space .00 KB

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
Memory Address 0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver
Memory Address 0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver
Memory Address 0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver

I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus
I/O Port 0x0000A000-0x0000AFFF PCI bus

Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge

Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF0000000-0xFFFFFFFF PCI bus
Memory Address 0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge

I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus
I/O Port 0x00000000-0x000002FF PCI bus

Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver
I/O Port 0x00002000-0x00002FFF PCI bus
I/O Port 0x00002000-0x00002FFF PCI bus

I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter

Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus
Memory Address 0xD8000000-0xDFFFFFFF PCI bus

Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

Memory Address 0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver
Memory Address 0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver
Memory Address 0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver

I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00009000-0x00009FFF PCI bus
I/O Port 0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter

PCI bus
HP Baseboard Management Controller Interface Driver

Memory Address 0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xF01C0000-0xF01FFFFF	Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus
Memory Address 0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xE0000000-0xEFFFFFFF PCI bus

Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus
Memory Address 0xD0000000-0xD7FFFFFF PCI bus

I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
I/O Port 0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

Memory Address 0xC8020000-0xC802FFFF Intel(R) PRO/1000 T Server Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter
Memory Address 0xC8020000-0xC802FFFF QLogic QLA23xx PCI Fibre Channel Adapter

Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus
Memory Address 0xC0000000-0xC7FFFFFF PCI bus

I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus
I/O Port 0x00001000-0x00001FFF PCI bus

Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport)

I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x00004000-0x00005FFF PCI bus
I/O Port 0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge

I/O Port 0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport)

I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge
I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge

I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge

I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge

I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge

I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)
I/O Port 0x0000C000-0x0000DFFF PCI bus
I/O Port 0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge

I/O Port 0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport)

I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus

I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter
I/O Port 0x00008000-0x00008FFF PCI bus
I/O Port 0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter

Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
Memory Address 0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)

Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Memory Address 0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)

[DMA]

Resource Device Status

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource Device Status
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000000-0x000002FF PCI bus OK
0x00000400-0x00000FFF PCI bus OK
0x00000D80-0x00000DBF HP MP Serial AUX/UPS Port (COM3) OK

0x00000D00-0x00000D7F HP Core-Io Device 01 OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00001000-0x00001FFF PCI bus OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0x00002000-0x00002FFF PCI bus OK
0x00002000-0x00002FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF LSI Logic 53C1010-66 Device OK

0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003000-0x00003FFF PCI bus OK
0x00003100-0x000031FF LSI Logic 53C1010-66 Device OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00004000-0x00005FFF PCI bus OK
0x00004000-0x00005FFF PCI standard PCI-to-PCI bridge OK

0x00004000-0x00005FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00006000-0x00007FFF PCI bus OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF Smart Array 6400 Controller (Non-Miniport) OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00008000-0x00008FFF PCI bus OK
0x00008000-0x00008FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x00009000-0x00009FFF PCI bus OK
0x00009000-0x00009FFF QLogic QLA23xx PCI Fibre Channel Adapter OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000A000-0x0000AFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000B000-0x0000BFFF PCI bus OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C000-0x0000DFFF PCI bus OK
0x0000C000-0x0000DFFF PCI standard PCI-to-PCI bridge OK

0x0000C000-0x0000DFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000C100-0x0000C1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E000-0x0000FFFF PCI bus OK
0x0000E000-0x0000FFFF PCI standard PCI-to-PCI bridge OK

0x0000E000-0x0000FFFF Smart Array 6400 Controller (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x0000E100-0x0000E1FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00004100-0x000041FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0x00006100-0x000061FF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

[IRQs]

Resource	Device	Status
IRQ 16	Microsoft ACPI-Compliant System	OK
IRQ 48	HP Core-Io Device 01	OK
IRQ 17	HP Baseboard Management Controller Interface Driver	OK
IRQ 35	Intel(R) PRO/1000 T Server Adapter	OK
IRQ 46	Broadcom NetXtreme Gigabit Ethernet	OK
IRQ 57	LSI Logic 53C1010-66 Device	OK
IRQ 58	LSI Logic 53C1010-66 Device	OK
IRQ 68	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 79	Emulex LightPulse LP1050, Storport Miniport Driver	OK
IRQ 90	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 101	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 134	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 135	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 145	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 146	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 194	Emulex LightPulse LP1050, Storport Miniport Driver	OK
IRQ 216	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 217	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 227	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 228	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 238	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 249	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 282	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 283	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 293	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 294	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 364	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 365	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 375	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 376	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 386	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 397	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 430	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 431	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 441	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 442	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 512	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 513	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 523	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 524	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 534	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 545	QLogic QLA23xx PCI Fibre Channel Adapter	OK
IRQ 578	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 579	Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)	OK
IRQ 589	Smart Array 6400 Controller (Non-Miniport)	OK

IRQ 590 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 638 Emulex LightPulse LP1050, Storport Miniport Driver OK

IRQ 660 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 661 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 671 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 672 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 682 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 693 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 726 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 727 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 737 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 738 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 786 Emulex LightPulse LP1050, Storport Miniport Driver OK

IRQ 808 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 809 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 819 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 820 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 830 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 841 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 874 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 875 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 885 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 886 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 956 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 957 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 967 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 968 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 978 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 989 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 1022 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1023 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 1033 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1034 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 1104 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1105 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 1115 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1116 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 1126 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 1137 QLogic QLA23xx PCI Fibre Channel Adapter OK

IRQ 1170 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1171 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
IRQ 1181 Smart Array 6400 Controller (Non-Miniport) OK

IRQ 1182 Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

[Memory]

Resource	Device	Status
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0000000-0xC7FFFFFF	PCI bus	OK
0xC0040000-0xC0040FFF	HP MP Serial AUX/UPS Port (COM3)	OK
0xC0041000-0xC00413FF	HP Core-Io Device 01	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	Intel(R) PRO/1000 T Server Adapter	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8000000-0xCFFFFFFF	PCI bus	OK
0xC8020000-0xC802FFFF	Intel(R) PRO/1000 T Server Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xC8020000-0xC802FFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK
0xD0000000-0xD7FFFFFF	PCI bus	OK

0xD0000000-0xD7FFFFFF PCI bus OK
0xD0010000-0xD001FFFF Broadcom NetXtreme Gigabit Ethernet OK

0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8000000-0xDFFFFFFF PCI bus OK
0xD8044000-0xD80443FF LSI Logic 53C1010-66 Device OK

0xD8040000-0xD8041FFF LSI Logic 53C1010-66 Device OK

0xD8045000-0xD80453FF LSI Logic 53C1010-66 Device OK

0xD8042000-0xD8043FFF LSI Logic 53C1010-66 Device OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK

0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK
0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0000000-0xEFFFFFFF PCI bus OK
0xE0000000-0xEFFFFFFF PCI standard PCI-to-PCI bridge OK

0xE0080000-0xE0081FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0040000-0xE007FFFF Smart Array 6400 Controller (Non-Miniport)
OK
0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK

0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK
0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0000000-0xFFFFFFFF PCI bus OK
0xF0000000-0xFFFFFFFF PCI standard PCI-to-PCI bridge OK

0xF0040000-0xF0040FFF Emulex LightPulse LP1050, Storport Miniport
Driver OK
0xF0041000-0xF00410FF Emulex LightPulse LP1050, Storport Miniport
Driver OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xC0020000-0xC0020FFF QLogic QLA23xx PCI Fibre Channel Adapter
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport)
OK

0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0100000-0xE0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE0102000-0xE0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0100000-0xF0101FFF Smart Array 6400 Controller (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF0102000-0xF0103FFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xD0040000-0xD0040FFF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xD0041000-0xD00410FF Emulex LightPulse LP1050, Storport Miniport Driver OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE01C0000-0xE01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xE00C0000-0xE00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF01C0000-0xF01FFFFF Smart Array 6400 Controller (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK
0xF00C0000-0xF00FFFFF Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version
	Size	Creation Date			
c:\win1828\system32\msadp32.acm	Microsoft Corporation				
	OK	C:\WIN1828\system32\MSADP32.ACM			
	5.2.3790.0 (srv03_rtm.030324-2048)	49.00 KB (50,176 bytes)	3/30/2005 3:20 PM		
c:\win1828\system32\msgsm32.acm	Microsoft Corporation				
	OK	C:\WIN1828\system32\MSGSM32.ACM			
	5.2.3790.0 (srv03_rtm.030324-2048)	66.50 KB (68,096 bytes)	3/30/2005 3:20 PM		
c:\win1828\system32\msg711.acm	Microsoft Corporation				
	OK	C:\WIN1828\system32\MSG711.ACM			
	5.2.3790.0 (srv03_rtm.030324-2048)	33.00 KB (33,792 bytes)	3/30/2005 3:20 PM		
c:\win1828\system32\imaadp32.acm	Microsoft Corporation				
	OK	C:\WIN1828\system32\IMAADP32.ACM			
	5.2.3790.0 (srv03_rtm.030324-2048)	55.00 KB (56,320 bytes)	3/30/2005 3:20 PM		
c:\win1828\system32\tssoft32.acm	DSP GROUP, INC.				
	OK	C:\WIN1828\system32\TSSOFT32.ACM	1.01		
	29.00 KB (29,696 bytes)	3/30/2005 3:21 PM			

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version
	Size	Creation Date			
c:\win1828\system32\msrle32.dll	Microsoft Corporation				
	OK	C:\WIN1828\system32\MSRLE32.DLL			
	5.2.3790.0 (srv03_rtm.030324-2048)	24.50 KB (25,088 bytes)	3/30/2005 3:20 PM		
c:\win1828\system32\msvidc32.dll	Microsoft Corporation				
	OK	C:\WIN1828\system32\MSVIDC32.DLL			
	5.2.3790.0 (srv03_rtm.030324-2048)	67.00 KB (68,608 bytes)	3/30/2005 3:20 PM		

[CD-ROM]

Item	Value

[Sound Device]

Item	Value

[Display]

Item	Value

[Infrared]

Item	Value

[Input]

[Keyboard]

Item	Value

[Pointing Device]

Item	Value

[Modem]

Item	Value

[Network]

[Adapter]

Item	Value
Name	[00000001] Intel(R) PRO/1000 T Server Adapter
Adapter Type	Ethernet 802.3
Product Type	Intel(R) PRO/1000 T Server Adapter
Installed	Yes

PNP Device ID
PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\5&2A6
94D11&0&08
Last Reset 11/21/2005 2:08 PM
Index 1
Service Name E1000
IP Address 15.1.101.78
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server 15.1.101.1
DHCP Lease Expires 11/29/2005 2:34 PM
DHCP Lease Obtained 11/21/2005 2:34 PM
MAC Address 00:03:47:72:A1:BE
Memory Address 0xC8000000-0xCFFFFFFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 35
Driver c:\win1828\system32\drivers\e1000645.sys (7.2.17.5 built by:
WinDDK, 410.00 KB (419,840 bytes), 3/30/2005 9:07 AM)

Name [00000002] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID
PCI\VEN_14E4&DEV_1645&SUBSYS_128A103C&REV_15\5&38E
DBD97&0&08
Last Reset 11/21/2005 2:08 PM
Index 2
Service Name b57nd
IP Address 10.193.25.78
IP Subnet 255.255.254.0
Default IP Gateway 10.193.24.1
DHCP Enabled Yes
DHCP Server 10.193.8.64
DHCP Lease Expires 11/29/2005 2:34 PM
DHCP Lease Obtained 11/21/2005 2:34 PM
MAC Address 00:04:76:DF:CE:E3
Memory Address 0xD0010000-0xD001FFFF
IRQ Channel IRQ 46
Driver c:\win1828\system32\drivers\b57xp64.sys (2.91.0.0 built by:
WinDDK, 499.25 KB (511,232 bytes), 3/30/2005 9:07 AM)

Name [00000003] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 3
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000004] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 11/21/2005 2:08 PM
Index 4
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000005] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 11/21/2005 2:08 PM
Index 5
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30

Name [00000006] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 11/21/2005 2:08 PM
Index 6
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30

Name [00000007] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 11/21/2005 2:08 PM
Index 7
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000008] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 11/21/2005 2:08 PM
Index 8
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000009] Intel(R) PRO/1000 MT Dual Port Network
Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 9
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000010] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 10
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000011] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 11
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000012] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 12
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Lease Expires Not Available
DHCP Server Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000013] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 13
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000014] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 14
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000015] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 15
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Lease Expires Not Available
DHCP Server Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000016] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 16
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000017] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 17
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000018] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 18
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No

DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000019] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 19
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000020] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 20
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000021] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 21
Service Name E1000
IP Address Not Available
IP Subnet Not Available
DHCP Enabled No
Default IP Gateway Not Available
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000022] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 22
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000023] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 23
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000024] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 24
Service Name E1000
IP Address Not Available
IP Subnet Not Available
DHCP Enabled No
Default IP Gateway Not Available
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000025] Intel(R) PRO/1000 MT Dual Port Server Adapter

Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 25
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000026] Intel(R) PRO/1000 MT Dual Port Server Adapter

Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 26
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000027] Intel(R) PRO/1000 MT Dual Port Server Adapter

Adapter Type Not Available
Product Type Intel(R) PRO/1000 MT Dual Port Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 27
Service Name E1000
IP Address Not Available
IP Subnet Not Available

Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000028] Intel(R) PRO/1000 MT Dual Port Server Adapter

Adapter TypeNot Available
Product TypeIntel(R) PRO/1000 MT Dual Port Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 28
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000029] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter TypeNot Available
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 29
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000030] Intel(R) PRO/1000 MT Dual Port Network Connection
Adapter TypeNot Available
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID Not Available
Last Reset 11/21/2005 2:08 PM
Index 30
Service Name E1000
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD Tcpip [UDP/IP]
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP UDP Service Provider
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 16 bytes
Maximum Message Size 63.93 KB (65,467 bytes)
Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\win1828\system32\wsock32.dll
Size 23.00 KB (23,552 bytes)
Version 5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item Value

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 16.34 GB (17,544,654,848 bytes)
Free Space 4.34 GB (4,660,953,088 bytes)
Volume Name Syspart
Volume Serial Number 3C494C82

Drive G:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 33.91 GB (36,413,280,256 bytes)
Free Space 12.16 GB (13,061,488,640 bytes)
Volume Name G_Drive
Volume Serial Number D0053378

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive O:
Description Network Connection
Provider Name \\orcamaster\d$

Drive S:
Description Network Connection
Provider Name \\hpwieshare\d$

Drive X:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 457.91 GB (491,681,476,608 bytes)
Free Space 446.83 GB (479,779,086,336 bytes)
Volume Name Log_Backup
Volume Serial Number F0F1EA47

[Disks]

Item Value
Description \\.\PHYSICALDRIVE99
Manufacturer Not Available
Model Not Available
Media Loaded Yes
Bytes/Sector 512
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255

Description \\.\PHYSICALDRIVE100
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255

Description \\.\PHYSICALDRIVE11
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #11, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #11, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE12
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #12, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #12, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #12, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE91
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #91, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #91, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE92
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #92, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #92, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #92, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #92, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #92, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE119
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #119, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #119, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE120
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #120, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #120, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #120, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE113
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #113, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #113, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE114
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #114, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #114, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #114, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE41
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #41, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #41, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE42

Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #42, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE97
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #97, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #97, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE98
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #98, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE109
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #109, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #109, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE110
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #110, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #110, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #110, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE51
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #51, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #51, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE52
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #52, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Partition Disk #52, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #52, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #52, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #52, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE21
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #21, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #21, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE22
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #22, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE33
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #33, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #33, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE34
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #34, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE43
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #43, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #43, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE44
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #44, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #44, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #44, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #44, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #44, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE49
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #49, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #49, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE50
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #50, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE35
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #35, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #35, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE36
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #36, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #36, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #36, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #36, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #36, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE71
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #71, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #71, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE72
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #72, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #72, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #72, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #72, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #72, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE15

Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #15, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE16
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #16, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #16, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE5
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #5, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE6
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #6, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE69
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #69, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #69, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE70
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #70, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE63
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #63, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes

Partition Disk #63, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE64
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #64, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #64, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #64, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #64, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #64, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE67
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #67, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #67, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE68
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #68, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #68, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #68, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #68, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #68, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE65
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #65, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #65, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE66
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #66, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE3
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #3, Partition #1

Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE4
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #4, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #4, Partition #1
Partition Size 457.91 GB (491,681,480,704 bytes)
Partition Starting Offset 313,560,924,160 bytes

Description \\.\PHYSICALDRIVE17
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #17, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE18
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #18, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE37
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #37, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #37, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE38
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #38, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE89
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #89, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #89, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE90
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #90, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)

Partition Starting Offset 136,314,880 bytes
Partition Disk #90, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #90, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #90, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #90, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE105
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #105, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #105, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE106
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #106, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE73
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #73, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #73, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE74
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #74, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE47
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #47, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #47, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE48
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #48, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #48, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #48, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #48, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #48, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE45
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #45, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #45, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE46
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #46, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE31
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
SCSI Bus Not Available
Partitions 2
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #31, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #31, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE32
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #32, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #32, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #32, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE85
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #85, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #85, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE86
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #86, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #86, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #86, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #86, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #86, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE19
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk

Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #19, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #19, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE20
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #20, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #20, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #20, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE107
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #107, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #107, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE108
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #108, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #108, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #108, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE53
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #53, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #53, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE54
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #54, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE95
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #95, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)

Partition Starting Offset 136,314,880 bytes
Partition Disk #95, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE96
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #96, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #96, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #96, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #96, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #96, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE121
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #121, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #121, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE122
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #122, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE23
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #23, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #23, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE24
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #24, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #24, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #24, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE39
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #39, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #39, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE40
Manufacturer Not Available

Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #40, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #40, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #40, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #40, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #40, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE111
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #111, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #111, Partition #1
Partition Starting Offset 118,712,434,688 bytes
Partition Size 61.20 GB (65,716,355,072 bytes)

Description \\.\PHYSICALDRIVE112
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #112, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #112, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #112, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE13
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #13, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #13, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE14
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #14, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE101
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #101, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #101, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE102
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available

SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #102, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #102, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #102, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #102, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #102, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE29
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #29, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #29, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE30
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #30, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE25
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #25, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #25, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE26
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE117
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #117, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #117, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE118
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #118, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #118, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)

Partition Starting Offset 313,560,924,160 bytes
Partition Disk #118, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE61
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #61, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #61, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE62
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #62, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #62, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #62, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #62, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE7
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #7, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #7, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE8
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #8, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #8, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE57
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #57, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #57, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE58
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #58, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE79
Manufacturer Not Available

Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #79, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #79, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE80
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #80, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #80, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #80, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #80, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #80, Partition #4
Partition Starting Offset 616,429,518,848 bytes
Partition Size 61.20 GB (65,716,355,072 bytes)

Description \\.\PHYSICALDRIVE115
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #115, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #115, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE116
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #116, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #116, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #116, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Manufacturer Not Available
Description \\.\PHYSICALDRIVE75
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #75, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #75, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE76
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #76, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #76, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #76, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #76, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #76, Partition #4

Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE103
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #103, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #103, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE104
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #104, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #104, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #104, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #104, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 549,853,399,040 bytes
Partition Disk #104, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE87
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #87, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #87, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE88
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #88, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #88, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #88, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #88, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 549,853,399,040 bytes
Partition Disk #88, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE93
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #93, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #93, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE94
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #94, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #94, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #94, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)

Partition Starting Offset 432,137,043,968 bytes
Partition Disk #94, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #94, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE77
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #77, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #77, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE78
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #78, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE81
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #81, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #81, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE82
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #82, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE27
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Port Not Available
SCSI Logical Unit Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #27, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #27, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE28
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #28, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #28, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #28, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes

Description \\.\PHYSICALDRIVE0
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available

Sectors/Track 63
Size 16.95 GB (18,202,544,640 bytes)
Total Cylinders 2,213
Total Sectors 35,551,845
Total Tracks 564,315
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 100.00 MB (104,857,600 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #0, Partition #1
Partition Size 400.00 MB (419,430,400 bytes)
Partition Starting Offset 104,875,008 bytes
Partition Disk #0, Partition #2
Partition Size 16.34 GB (17,544,657,408 bytes)
Partition Starting Offset 658,523,136 bytes

Description \\.\PHYSICALDRIVE1
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 16.95 GB (18,202,544,640 bytes)
Total Cylinders 2,213
Total Sectors 35,551,845
Total Tracks 564,315
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 172.54 MB (180,923,904 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #1, Partition #1
Partition Size 16.66 GB (17,889,984,000 bytes)
Partition Starting Offset 312,560,640 bytes

Description \\.\PHYSICALDRIVE2
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 33.91 GB (36,413,314,560 bytes)
Total Cylinders 4,427
Total Sectors 71,119,755
Total Tracks 1,128,885
Tracks/Cylinder 255
Partition Disk #2, Partition #0
Partition Size 33.91 GB (36,413,282,304 bytes)
Partition Starting Offset 32,256 bytes

Description \\.\PHYSICALDRIVE55
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #55, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #55, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE56
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #56, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #56, Partition #1
Partition Starting Offset 313,560,924,160 bytes
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Disk #56, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #56, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #56, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE9
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #9, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #9, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE10
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 292.03 GB (313,564,124,160 bytes)
Total Cylinders 38,122
Total Sectors 612,429,930
Total Tracks 9,721,110
Tracks/Cylinder 255
Partition Disk #10, Partition #0

Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes

Description \\.\PHYSICALDRIVE83
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #83, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #83, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE84
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #84, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #84, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Disk #84, Partition #2
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #84, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description \\.\PHYSICALDRIVE59
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 171.87 GB (184,542,382,080 bytes)
Total Cylinders 22,436
Total Sectors 360,434,340
Total Tracks 5,721,180
Tracks/Cylinder 255
Partition Disk #59, Partition #0
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #59, Partition #1
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 118,712,434,688 bytes

Description \\.\PHYSICALDRIVE60
Manufacturer Not Available
Model Not Available
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 5
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 749.94 GB (805,238,461,440 bytes)
Total Cylinders 97,898
Total Sectors 1,572,731,370
Total Tracks 24,963,990
Tracks/Cylinder 255
Partition Disk #60, Partition #0
Partition Size 291.90 GB (313,424,609,280 bytes)
Partition Starting Offset 136,314,880 bytes
Partition Disk #60, Partition #1
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 313,560,924,160 bytes
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 432,137,043,968 bytes
Partition Disk #60, Partition #3
Partition Size 110.43 GB (118,576,119,808 bytes)
Partition Starting Offset 497,853,399,040 bytes
Partition Disk #60, Partition #4
Partition Size 61.20 GB (65,716,355,072 bytes)
Partition Starting Offset 616,429,518,848 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 3
SCSI Target ID 0
Sectors/Track 63
Size 957.12 GB (1,027,699,384,320 bytes)
Total Cylinders 124,944
Total Sectors 2,007,225,360
Total Tracks 31,860,720
Tracks/Cylinder 255
Partition Disk #126, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #126, Partition #1
Partition Size 957.00 GB (1,027,571,276,288 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 1
SCSI Target ID 0
Sectors/Track 63
Size 957.12 GB (1,027,699,384,320 bytes)
Total Cylinders 124,944
Total Sectors 2,007,225,360
Total Tracks 31,860,720
Tracks/Cylinder 255
Partition Disk #124, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)

Partition Starting Offset 17,408 bytes
Partition Disk #124, Partition #1
Partition Size 957.00 GB (1,027,571,276,288 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 2
SCSI Target ID 0
Sectors/Track 63
Size 957.12 GB (1,027,699,384,320 bytes)
Total Cylinders 124,944
Total Sectors 2,007,225,360
Total Tracks 31,860,720
Tracks/Cylinder 255
Partition Disk #125, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #125, Partition #1
Partition Size 957.00 GB (1,027,571,276,288 bytes)
Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model COMPAQ MSA1000 VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 0
SCSI Target ID 0
Sectors/Track 63
Size 957.12 GB (1,027,699,384,320 bytes)
Total Cylinders 124,944
Total Sectors 2,007,225,360
Total Tracks 31,860,720
Tracks/Cylinder 255
Partition Disk #123, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #123, Partition #1
Partition Size 957.00 GB (1,027,571,276,288 bytes)
Partition Starting Offset 134,235,136 bytes

[SCSI]

Item Value
Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID
PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\5&137A5C4D&0&08
I/O Port 0x00003000-0x00003FFF
Memory Address 0xD8044000-0xD80443FF
Memory Address 0xD8040000-0xD8041FFF
IRQ Channel IRQ 57
Driver c:\win1828\system32\drivers\sym_u3.sys (5.09.06.00 (NT.041001-1408), 73.00 KB (74,752 bytes), 3/30/2005 3:21 PM)

Name LSI Logic 53C1010-66 Device
Manufacturer LSI Logic Inc.
Status OK
PNP Device ID
PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\5&137A5C4D&0&09
I/O Port 0x00003100-0x000031FF
Memory Address 0xD8045000-0xD80453FF
Memory Address 0xD8042000-0xD8043FFF
IRQ Channel IRQ 58
Driver c:\win1828\system32\drivers\sym_u3.sys (5.09.06.00 (NT.041001-1408), 73.00 KB (74,752 bytes), 3/30/2005 3:21 PM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&266093A2&0&2008
Memory Address 0xE0080000-0xE0081FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE0040000-0xE007FFFF
IRQ Channel IRQ 68
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Emulex LightPulse LP1050, Storport Miniport Driver
Manufacturer Emulex
Status OK
PNP Device ID
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&32C61491&0&08
Memory Address 0xF0040000-0xF0040FFF
Memory Address 0xF0041000-0xF00410FF
I/O Port 0x00006000-0x00007FFF
IRQ Channel IRQ 79
Driver c:\win1828\system32\drivers\elxstor.sys (6-1.10A4 3/21/2005 WS2K3 64 bit IA64 built by: WinDDK, 690.00 KB (706,560 bytes), 9/29/2005 2:01 PM)
Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3633CCA9&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 90
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&251997C5&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 101
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1457B150&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
IRQ Channel IRQ 134
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1457B150&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
IRQ Channel IRQ 135
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)

Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&2B186594&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
IRQ Channel IRQ 145
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&2B186594&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
IRQ Channel IRQ 146
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Emulex LightPulse LP1050, Storport Miniport Driver
Manufacturer Emulex
Status OK
PNP Device ID
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&3815FDFD&0&08
Memory Address 0xD0040000-0xD0040FFF
Memory Address 0xD0041000-0xD00410FF
I/O Port 0x00002000-0x00002FFF
IRQ Channel IRQ 194
Driver c:\win1828\system32\drivers\elxstor.sys (6-1.10A4 3/21/2005 WS2K3 64 bit IA64 built by: WinDDK, 690.00 KB (706,560 bytes), 9/29/2005 2:01 PM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3533F53F&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 216
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3533F53F&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 217
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&16EA0F70&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 227
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&16EA0F70&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 228
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&355C0D0F&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 238
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&8F8ADE6&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 249
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1600BE56&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 282
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1600BE56&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 283
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&8A8E882&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 293
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard

Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&8A8E882&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 294
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&6AA0D5F&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 364
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&6AA0D5F&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 365
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3B7BF75A&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 375
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3B7BF75A&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 376
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&345102A3&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 386
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&370AF391&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 397
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&285861AA&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 430
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&285861AA&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 431
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&27EBA306&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 441
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&27EBA306&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 442
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3B940DB4&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 512
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard

Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3B940DB4&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 513
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&33C2A78D&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 523
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&33C2A78D&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 524
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&266723A4&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 534
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&23AD32B6&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 545
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&26156FBC&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 578
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&26156FBC&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 579
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1B9584B9&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 589
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1B9584B9&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 590
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Emulex LightPulse LP1050, Storport Miniport Driver
Manufacturer Emulex
Status OK
PNP Device ID
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&36A998EE&0&08
Memory Address 0xD0040000-0xD0040FFF
Memory Address 0xD0041000-0xD00410FF
I/O Port 0x00002000-0x00002FFF
IRQ Channel IRQ 638
Driver c:\win1828\system32\drivers\elxstor.sys (6-1.10A4 3/21/2005 WS2K3 64 bit IA64 built by: WinDDK, 690.00 KB (706,560 bytes), 9/29/2005 2:01 PM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&20DD8E1E&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 660
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&20DD8E1E&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 661
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK

PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&14635F0&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 671
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&14635F0&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 672
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&7AB2207&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 682
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&A6512F5&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 693
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK

PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13220302&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 726
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13220302&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 727
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1EB0D7F0&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 737
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1EB0D7F0&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 738
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Emulex LightPulse LP1050, Storport Miniport Driver
Manufacturer Emulex
Status OK

PNP Device ID
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&24E12DE3&0&08
Memory Address 0xD0040000-0xD0040FFF
Memory Address 0xD0041000-0xD00410FF
I/O Port 0x00002000-0x00002FFF
IRQ Channel IRQ 786
Driver c:\win1828\system32\drivers\elxstor.sys (6-1.10A4 3/21/2005 WS2K3 64 bit IA64 built by: WinDDK, 690.00 KB (706,560 bytes), 9/29/2005 2:01 PM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&FD96073&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 808
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&FD96073&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 809
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&466FDC1&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 819
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard

Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&466FDC1&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 820
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&279B1EA1&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 830
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&22DA925C&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 841
Driver c:\win1828\system32\drivers\ql2300.sys (8.2.2.10 (W64 VI), 690.75 KB (707,328 bytes), 3/16/2005 8:45 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&F78FC9&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 874
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&F78FC9&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 875
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&36DED50F&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 885
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&36DED50F&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 886
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&273093EA&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 956
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&273093EA&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 957
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&301D439F&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 967
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&301D439F&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 968
Driver c:\win1828\system32\drivers\hpqcissb.sys (5.15.64.64 built by: WinDDK, 113.00 KB (115,712 bytes), 3/8/2005 10:49 AM)

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&331D07A6&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 978

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&B37B34F&0&08
I/O Port 0x00009000-0x00009FFF

Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 989

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&348D8DDD&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 1022

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&348D8DDD&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 1023

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&214639D8&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 1033

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&214639D8&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 1034

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1DDED019&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x00004000-0x00005FFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 1104

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1DDED019&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x00004100-0x000041FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 1105

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13DF1870&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x00006000-0x00007FFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 1115

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13DF1870&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x00006100-0x000061FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 1116

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3690080C&0&08
I/O Port 0x00008000-0x00008FFF
Memory Address 0xC0020000-0xC0020FFF
IRQ Channel IRQ 1126

Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3949F8FA&0&08
I/O Port 0x00009000-0x00009FFF
Memory Address 0xC8020000-0xC802FFFF
IRQ Channel IRQ 1137

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&FCF5F0&0&2008
Memory Address 0xE0100000-0xE0101FFF
I/O Port 0x0000C000-0x0000DFFF
Memory Address 0xE01C0000-0xE01FFFFF
IRQ Channel IRQ 1170

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&FCF5F0&0&2808
Memory Address 0xE0102000-0xE0103FFF
I/O Port 0x0000C100-0x0000C1FF
Memory Address 0xE00C0000-0xE00FFFFF
IRQ Channel IRQ 1171

Name Smart Array 6400 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&19905E4A&0&2008
Memory Address 0xF0100000-0xF0101FFF
I/O Port 0x0000E000-0x0000FFFF
Memory Address 0xF01C0000-0xF01FFFFF
IRQ Channel IRQ 1181

Name Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&19905E4A&0&2808
Memory Address 0xF0102000-0xF0103FFF
I/O Port 0x0000E100-0x0000E1FF
Memory Address 0xF00C0000-0xF00FFFFF
IRQ Channel IRQ 1182

[IDE]

Item Value

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code

Not Available ACPI\HWP1000\A56E04661474 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662909 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661350 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620989 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627456 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627455 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662630 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662881 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662926 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661423 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04587395 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627462 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662810 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661477 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661487 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662609 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620957 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662929 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04739725 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620738 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627511 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661471 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661512 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639005 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661359 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620821 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639159 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639130 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620833 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662841 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620702 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627436 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620635 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662570 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627572 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620643 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627576 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662825 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662700 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620585 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620709 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620770 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661341 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662815 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639045 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661336 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661533 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639154 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661520 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662567 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627429 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627469 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662945 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627480 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04627405 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04639122 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620931 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620644 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662816 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04538867 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620636 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04662631 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04620932 The drivers for this device are not installed.
Not Available ACPI\HWP1000\A56E04661421 The drivers for this device are not installed.

[USB]

Device PNP Device ID

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start
Mode	State	Status	Error Control	Accept Pause	Accept Stop
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Ignore	No	No

```
acpi        Microsoft ACPI Driver
            c:\win1828\system32\drivers\acpi.sys    Kernel Driver Yes
            Boot        Running     OK          Normal      No
            Yes
acpiec      ACPIEC      c:\win1828\system32\drivers\acpiec.sys  Kernel
Driver      No          Disabled    Stopped     OK          Normal
            No          No
adpu160m    adpu160m    Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

adpu320     adpu320     Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

afcnt       afcnt       Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

afd         AFD         c:\win1828\system32\drivers\afd.sys     Kernel
Driver      Yes         System      Running     OK          Normal
            No          Yes
aic78u2     aic78u2     Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

aic78xx     aic78xx     Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

aliide      AliIde      Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

arc         arc         Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

asyncmac    RAS Asynchronous Media Driver
            c:\win1828\system32\drivers\asyncmac.sys    Kernel
Driver      No          Manual      Stopped     OK          Normal
            No          No
atapi       atapi       c:\win1828\system32\drivers\atapi.sys   Kernel
Driver      No          Disabled    Stopped     OK          Normal
            No          No
atdisk      Atdisk      Not Available Kernel Driver No          Disabled
            Stopped     OK          Ignore      No          No

atmarpc     ATM ARP Client Protocol
            c:\win1828\system32\drivers\atmarpc.sys     Kernel
Driver      No          Manual      Stopped     OK          Normal
            No          No
audstub     Audio Stub Driver
            c:\win1828\system32\drivers\audstub.sys     Kernel
Driver      Yes         Manual      Running     OK          Normal
            No          Yes

b57nd       Broadcom NetXtreme Gigabit Ethernet
            c:\win1828\system32\drivers\b57xp64.sys     Kernel
Driver      Yes         Manual      Running     OK          Normal
beep        Beep        c:\win1828\system32\drivers\beep.sys    Kernel
Driver      Yes         System      Running     OK          Normal
            No          Yes
cbidf2k     cbidf2k     c:\win1828\system32\drivers\cbidf2k.sys
            Kernel Driver No        Disabled    Stopped     OK
            Normal      No          No
cdfs        Cdfs        c:\win1828\system32\drivers\cdfs.sys    File
System Driver           No          Disabled    Stopped     OK
            Normal      No          No
cdrom       Cdrom       c:\win1828\system32\drivers\cdrom.sys   Kernel
Driver      No          System      Stopped     OK          Ignore
            No          No
changer     Changer     Not Available Kernel Driver No          System
            Stopped     OK          Ignore      No          No

clusdisk    Cluster Disk Driver
            c:\win1828\system32\drivers\clusdisk.sys    Kernel
Driver      No          Disabled    Stopped     OK          Normal
            No          No
cmdide      CmdIde      Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

cpqarry2    cpqarry2    Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

cpqcisse    CPQCISSE    c:\win1828\system32\drivers\cpqcisse.sys
            Kernel Driver No        Manual      Stopped     OK
            Normal      No          No
cpqcissm    cpqcissm    c:\win1828\system32\drivers\cpqcissm.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
cpqfcac     CPQFCAC     c:\win1828\system32\drivers\cpqfcac.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
cpqfcalm    cpqfcalm    Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

crcdisk     CRC Disk Filter Driver
            c:\win1828\system32\drivers\crcdisk.sysKernel Driver Yes
            Boot        Running     OK          Normal      No
            Yes
dfsdriver   DfsDriver    c:\win1828\system32\drivers\dfs.sys     File
System Driver           Yes         Boot        Running     OK
            Normal      No          Yes
disk        Disk Driver c:\win1828\system32\drivers\disk.sys    Kernel
Driver      Yes         Boot        Running     OK          Normal
            No          Yes
dmboot      dmboot      c:\win1828\system32\drivers\dmboot.sys
            Kernel Driver Yes       Disabled    Running     OK
            Normal      No          Yes

dmio        Logical Disk Manager Driver
            c:\win1828\system32\drivers\dmio.sys    Kernel Driver Yes
            Boot        Running     OK          Normal      No
dmload      dmload      c:\win1828\system32\drivers\dmload.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
dpti2o      dpti2o      Not Available Kernel Driver No          Disabled
            Stopped     OK          Normal      No          No

e1000       Intel(R) PRO/1000 Adapter Driver
            c:\win1828\system32\drivers\e1000645.sys    Kernel
Driver      Yes         Manual      Running     OK          Normal
            No          Yes
elxadjct    elxadjct    c:\win1828\system32\drivers\elxadjct.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
elxstor     elxstor     c:\win1828\system32\drivers\elxstor.sys Kernel
Driver      Yes         Boot        Running     OK          Normal
            No          Yes
fastfat     Fastfat     c:\win1828\system32\drivers\fastfat.sys File
System Driver           No          Disabled    Stopped     OK
            Normal      No          No
fdc         Fdc         c:\win1828\system32\drivers\fdc.sys     Kernel
Driver      No          System      Stopped     OK          Ignore
            No          No
filemon     FILEMON     \??\c:\win1828\system32\drivers\filem.sys
            Kernel Driver No        Manual      Stopped     OK
            Normal      No          No
fips        Fips        c:\win1828\system32\drivers\fips.sys    Kernel
Driver      Yes         System      Running     OK          Normal
            No          Yes
flpydisk    Flpydisk    c:\win1828\system32\drivers\flpydisk.sys
            Kernel Driver No        System      Stopped     OK
            Ignore      No          No
fltmgr      FltMgr      c:\win1828\system32\drivers\fltmgr.sys  File
System Driver           Yes         Boot        Running     OK
            Normal      No          Yes
ftdisk      Volume Manager Driver
            c:\win1828\system32\drivers\ftdisk.sys  Kernel Driver Yes
            Boot        Running     OK          Normal      No
            Yes
gpc         Generic Packet Classifier
            c:\win1828\system32\drivers\msgpc.sys   Kernel Driver Yes
            Manual      Running     OK          Normal      No
            Yes
hpcisss     hpcisss     c:\win1828\system32\drivers\hpcisss.sys
            Kernel Driver Yes       Boot        Running     OK
            Normal      No          Yes
hpcsr       HP CSR OpRegion Driver
            c:\win1828\system32\drivers\hpcsr.sys   Kernel Driver Yes
            Boot        Running     OK          Normal      No
            Yes
```

```
hphlth          HP Baseboard Management Controller Interface Driver
                c:\win1828\system32\drivers\hphlth.sys Kernel Driver Yes
                Manual     Running    OK         Normal     No
                Yes
hpmpser         HP MP Driver c:\win1828\system32\drivers\hpmpser.sys
                Kernel Driver No         Manual     Stopped    OK
                Normal     No         No
hpn             hpn        Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

hpqcissb        Smart Array Controllers Non-Miniport Bus Driver
                c:\win1828\system32\drivers\hpqcissb.sys          Kernel
Driver          Yes        Boot       Running    OK         Normal
                No         Yes
hpqcissd        Smart Array Controllers Non-Miniport Disk Driver
                c:\win1828\system32\drivers\hpqcissd.sys          Kernel
Driver          Yes        Boot       Running    OK         Normal
                No         Yes
http            HTTP       c:\win1828\system32\drivers\http.sys  Kernel
Driver          No         Manual     Stopped    OK         Normal
                No         No
i2omgmt         i2omgmt    Not Available Kernel Driver No         System
                Stopped    OK         Normal     No         No

iirsp           iirsp      Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

intelide        IntelIde   Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

ip6fw           IPv6 Windows Firewall Driver
                c:\win1828\system32\drivers\ip6fw.sys  Kernel Driver No
                Manual     Stopped    OK         Normal     No
                No
ipfilterdriver  IP Traffic Filter Driver
                c:\win1828\system32\drivers\ipfltdrv.sysKernel Driver No
                Manual     Stopped    OK         Normal     No
                No
ipinip          IP in IP Tunnel Driver
                c:\win1828\system32\drivers\ipinip.sys  Kernel Driver No
                Manual     Stopped    OK         Normal     No
                No
ipnat           IP Network Address Translator
                c:\win1828\system32\drivers\ipnat.sys  Kernel Driver Yes
                Manual     Running    OK         Normal     No
                Yes
ipsec           IPSEC driver c:\win1828\system32\drivers\ipsec.sys  Kernel
Driver          Yes        System     Running    OK         Normal
                No         Yes
isapnp          isapnp     Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

kbdclass        Keyboard Class Driver
                c:\win1828\system32\drivers\kbdclass.sys          Kernel
Driver          Yes        System     Running    OK         Normal
kldbgdrv        kldbgdrv   Yes        c:\win1828\system32\kldbgdrv.sys  Kernel
Driver          No         Manual     Stopped    OK         Normal
                No         No
ksecdd          KSecDD     c:\win1828\system32\drivers\ksecdd.sysKernel
Driver          Yes        Boot       Running    OK         Normal
                No         Yes
ksthunk         Kernel Streaming WOW64 Thunk Service
                c:\win1828\system32\drivers\ksthunk.sys          Kernel
Driver          Yes        Manual     Running    OK         Normal
                No         Yes
lp6nds35        lp6nds35   c:\win1828\system32\drivers\lp6nds35.sys
                Kernel Driver Yes        Boot       Running    OK
                Normal     No         Yes
mnmdd           mnmdd      Not Available Kernel Driver No         System
                Stopped    OK         Ignore     No         No

modem           Modem      c:\win1828\system32\drivers\modem.sys
                Kernel Driver No         Manual     Stopped    OK
                Ignore     No         No
mouclass        Mouse Class Driver
                c:\win1828\system32\drivers\mouclass.sys          Kernel
Driver          Yes        System     Running    OK         Normal
                No         Yes
mountmgr        Mount Point Manager
                c:\win1828\system32\drivers\mountmgr.sys          Kernel
Driver          Yes        Boot       Running    OK         Normal
                No         Yes
mraid35x        mraid35x   Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

mrxdav          WebDav Client Redirector
                c:\win1828\system32\drivers\mrxdav.sys           File
System Driver              No         Manual     Stopped    OK
                Normal     No         No
mrxsmb          MRXSMB     c:\win1828\system32\drivers\mrxsmb.sys
                File System Driver    Yes        System     Running
                OK         Normal     No         Yes
msfs            Msfs       c:\win1828\system32\drivers\msfs.sys  File
System Driver              Yes        System     Running    OK
                Normal     No         Yes
mssmbios        Microsoft System Management BIOS Driver
                c:\win1828\system32\drivers\mssmbios.sys          Kernel
Driver          Yes        Manual     Running    OK         Normal
                No         Yes
mup             Mup        c:\win1828\system32\drivers\mup.sys   File
System Driver              Yes        Boot       Running    OK
                Normal     No         Yes

ndis            NDIS System Driver
                c:\win1828\system32\drivers\ndis.sys  Kernel Driver Yes
                Boot       Running    OK         Normal     No
ndistapi        Remote Access NDIS TAPI Driver
                c:\win1828\system32\drivers\ndistapi.sys          Kernel
Driver          Yes        Manual     Running    OK         Normal
                No         Yes
ndisuio         NDIS Usermode I/O Protocol
                c:\win1828\system32\drivers\ndisuio.sys          Kernel
Driver          No         Manual     Stopped    OK         Normal
                No         No
ndiswan         Remote Access NDIS WAN Driver
                c:\win1828\system32\drivers\ndiswan.sys          Kernel
Driver          Yes        Manual     Running    OK         Normal
                No         Yes
ndproxy         NDIS Proxy c:\win1828\system32\drivers\ndproxy.sys
                Kernel Driver Yes        Manual     Running    OK
                Normal     No         Yes
netbios         NetBIOS Interface
                c:\win1828\system32\drivers\netbios.sys          File
System Driver              Yes        System     Running    OK
                Normal     No         Yes
netbt           NetBios over Tcpip
                c:\win1828\system32\drivers\netbt.sys  Kernel Driver Yes
                System     Running    OK         Normal     No
                Yes
nfrd960         nfrd960    Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

npfs            Npfs       c:\win1828\system32\drivers\npfs.sys  File
System Driver              Yes        System     Running    OK
                Normal     No         Yes
ntfs            Ntfs       c:\win1828\system32\drivers\ntfs.sys  File
System Driver              Yes        Disabled   Running    OK
                Normal     No         Yes
null            Null       c:\win1828\system32\drivers\null.sys  Kernel
Driver          Yes        System     Running    OK         Normal
                No         Yes
partmgr         Partition Manager
                c:\win1828\system32\drivers\partmgr.sys          Kernel
Driver          Yes        Boot       Running    OK         Normal
                No         Yes
pci             PCI Bus Driver        c:\win1828\system32\drivers\pci.sys
                Kernel Driver Yes        Boot       Running    OK
                Critical   No         Yes
pciide          PCIIde     Not Available Kernel Driver No         Disabled
                Stopped    OK         Normal     No         No

pcmcia          Pcmcia     c:\win1828\system32\drivers\pcmcia.sysKernel
Driver          No         Disabled   Stopped    OK         Normal
                No         No
```

```
pdcomp      PDCOMP      Not Available Kernel Driver No          Manual
            Stopped     OK            Ignore        No          No

pdframe     PDFRAME     Not Available Kernel Driver No          Manual
            Stopped     OK            Ignore        No          No

pdreli      PDRELI      Not Available Kernel Driver No          Manual
            Stopped     OK            Ignore        No          No

pdrframe    PDRFRAME    Not Available Kernel Driver No          Manual
            Stopped     OK            Ignore        No          No

pptpminiport WAN Miniport (PPTP)
            c:\win1828\system32\drivers\raspptp.sys             Kernel
Driver      Yes         Manual        Running       OK          Normal
            No          Yes
processor   Processor Driver
            c:\win1828\system32\drivers\processr.sys            Kernel
Driver      Yes         Manual        Running       OK          Normal
            No          Yes
procinfo    ProcInfo    c:\win1828\system32\drivers\procinfo.sys
            Kernel Driver No          Manual        Stopped     OK
            Normal      No            No
ptilink     Direct Parallel Link Driver
            c:\win1828\system32\drivers\ptilink.sys Kernel Driver Yes
            Manual      Running       OK            Normal      No
            Yes
ql1080      ql1080      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

ql12160     ql12160     Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

ql1240      ql1240      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

ql1280      ql1280      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

ql2200      ql2200      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

ql2300      ql2300      c:\win1828\system32\drivers\ql2300.sys Kernel
Driver      Yes         Boot          Running       OK          Normal
            No          Yes
qlvika      qlvika      c:\win1828\system32\drivers\qlvika.sys Kernel
Driver      Yes         Auto          Running       OK          Normal
            No          Yes
rasacd      Remote Access Auto Connection Driver
            c:\win1828\system32\drivers\rasacd.sys Kernel Driver Yes
            System      Running       OK            Normal      No
            Yes

rasl2tp     WAN Miniport (L2TP)
            c:\win1828\system32\drivers\rasl2tp.sys Kernel Driver Yes
            Manual      Running       OK            Normal      No
raspppoe    Remote Access PPPOE Driver
            c:\win1828\system32\drivers\raspppoe.sys            Kernel
Driver      Yes         Manual        Running       OK          Normal
            No          Yes
raspti      Direct Parallel
            c:\win1828\system32\drivers\raspti.sys  Kernel Driver Yes
            Manual      Running       OK            Normal      No
            Yes
rdbss       Rdbss       c:\win1828\system32\drivers\rdbss.sys   File
System Driver           Yes           System        Running     OK
            Normal      No            Yes
rdpcdd      RDPCDD      c:\win1828\system32\drivers\rdpcdd.sys Kernel
Driver      Yes         System        Running       OK          Ignore
            No          Yes
rdpdr       Terminal Server Device Redirector Driver
            c:\win1828\system32\drivers\rdpdr.sys  Kernel Driver Yes
            Manual      Running       OK            Normal      No
            Yes
rdpwd       RDPWD       c:\win1828\system32\drivers\rdpwd.sys Kernel
Driver      Yes         Manual        Running       OK          Ignore
            No          Yes
serenum     Serenum Filter Driver
            c:\win1828\system32\drivers\serenum.sys             Kernel
Driver      No          Manual        Stopped       OK          Normal
            No          No
serial      Serial      c:\win1828\system32\drivers\serial.sys  Kernel
Driver      No          Auto          Stopped       OK          Ignore
            No          No
sfloppy     Sfloppy     c:\win1828\system32\drivers\sfloppy.sys
            Kernel Driver No          System        Stopped     OK
            Ignore      No            No
simbad      Simbad      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

slowmemory  Slow Memory Measurement
            c:\win1828\system32\drivers\slowmemory.sys          Kernel
Driver      No          Manual        Stopped       OK          Normal
            No          No
srv         Srv         c:\win1828\system32\drivers\srv.sys     File
System Driver           Yes           Manual        Running     OK
            Normal      No            Yes
swenum      Software Bus Driver
            c:\win1828\system32\drivers\swenum.sys              Kernel
Driver      Yes         Manual        Running       OK          Normal
            No          Yes
symc8xx     symc8xx     Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

symmpi      symmpi      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No
sym_hi      sym_hi      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

sym_u3      sym_u3      c:\win1828\system32\drivers\sym_u3.sys
            Kernel Driver Yes         Boot          Running     OK
            Normal      No            Yes
tcpip       TCP/IP Protocol Driver
            c:\win1828\system32\drivers\tcpip.sys  Kernel Driver Yes
            System      Running       OK            Normal      No
            Yes
tdpipe      TDPIPE      c:\win1828\system32\drivers\tdpipe.sys Kernel
Driver      No          Manual        Stopped       OK          Ignore
            No          No
tdtcp       TDTCP       c:\win1828\system32\drivers\tdtcp.sys   Kernel
Driver      Yes         Manual        Running       OK          Ignore
            No          Yes
termdd      Terminal Device Driver
            c:\win1828\system32\drivers\termdd.sys              Kernel
Driver      Yes         System        Running       OK          Normal
            No          Yes
toside      TosIde      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

udfs        Udfs        c:\win1828\system32\drivers\udfs.sys    File
System Driver           No            Disabled      Stopped     OK
            Normal      No            No
vgasave     VGA Display Controller.
            c:\win1828\system32\drivers\vga.sys    Kernel Driver No
            System      Stopped       OK            Ignore      No
            No
viaide      ViaIde      Not Available Kernel Driver No          Disabled
            Stopped     OK            Normal        No          No

volsnap     Storage volumes
            c:\win1828\system32\drivers\volsnap.sys             Kernel
Driver      Yes         Boot          Running       OK          Normal
            No          Yes
wanarp      Remote Access IP ARP Driver
            c:\win1828\system32\drivers\wanarp.sys              Kernel
Driver      Yes         Manual        Running       OK          Normal
            No          Yes
wdica       WDICA       Not Available Kernel Driver No          Manual
            Stopped     OK            Ignore        No          No

wlbs        Network Load Balancing
            c:\win1828\system32\drivers\wlbs.sys    Kernel Driver No
            Manual      Stopped       OK            Normal      No
            No

[Signed Drivers]
```

Device Name Signed Device Class Driver Version Driver Date Manufacturer INF Name Driver Name Device ID
Microsoft System Management BIOS Driver Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\SYSTEM\0001
Plug and Play Software Device Enumerator Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\SYSTEM\0000
Terminal Server Mouse Driver Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\RDP_MOU\0000
Terminal Server Keyboard Driver Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\RDP_KBD\0000
Terminal Server Device Redirector Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\RDPDR\0000
Direct Parallel Yes NET 5.2.3790.1828 10/1/2002 Microsoft netrasa.inf Not Available ROOT\MS_PTIMINIPORT\0000
WAN Miniport (PPTP) Yes NET 5.2.3790.1828 10/1/2002 Microsoft netrasa.inf Not Available ROOT\MS_PPTPMINIPORT\0000
WAN Miniport (PPPOE) Yes NET 5.2.3790.1828 10/1/2002 Microsoft netrasa.inf Not Available ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (IP) Yes NET 5.2.3790.1828 10/1/2002 Microsoft netrasa.inf Not Available ROOT\MS_NDISWANIP\0000
WAN Miniport (L2TP) Yes NET 5.2.3790.1828 10/1/2002 Microsoft netrasa.inf Not Available ROOT\MS_L2TPMINIPORT\0000
Video Codecs Yes MEDIA 5.2.3790.0 10/1/2002 (Standard system devices) wave.inf Not Available ROOT\MEDIA\MS_MMVID
Legacy Video Capture Devices Yes MEDIA 5.2.3790.0 10/1/2002 (Standard system devices) wave.inf Not Available ROOT\MEDIA\MS_MMVCD
Media Control Devices Yes MEDIA 5.2.3790.0 10/1/2002 (Standard system devices) wave.inf Not Available ROOT\MEDIA\MS_MMMCI
Legacy Audio Drivers Yes MEDIA 5.2.3790.0 10/1/2002 (Standard system devices) wave.inf Not Available ROOT\MEDIA\MS_MMDRV
Audio Codecs Yes MEDIA 5.2.3790.0 10/1/2002 (Standard system devices) wave.inf Not Available ROOT\MEDIA\MS_MMACM
Remote Access IP ARP Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_WANARP\0000
volsnap Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_VOLSNAP\0000
TDTCP Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_TDTCP\0000
TCP/IP Protocol Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_TCPIP\0000
Slow Memory Measurement Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_SLOWMEMORY\0000
sacdrv Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_SACDRV\0000
RDPWD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPWD\0000
RDPCDD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPCDD\0000
Remote Access Auto Connection Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RASACD\0000
qlvika Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_QLVIKA\0000
ProcInfo Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_PROCINFO\0000
Partition Manager Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_PARTMGR\0000
Null Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NULL\0000
NetBios over Tcpip Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NETBT\0000
NDProxy Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDPROXY\0000
NDIS Usermode I/O Protocol Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISUIO\0000
Remote Access NDIS TAPI Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISTAPI\0000
NDIS System Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDIS\0000
mountmgr Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_MOUNTMGR\0000
lp6nds35 Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_LP6NDS35\0000
ksecdd Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_KSECDD\0000
kldbgdrv Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_KLDBGDRV\0000
IPSEC driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_IPSEC\0000
IP Network Address Translator Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_IPNAT\0000
hpcisss Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_HPCISSS\0000
Generic Packet Classifier Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_GPC\0000
Fips Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_FIPS\0000
FILEMON Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_FILEMON\0000
dmload Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_DMLOAD\0000
dmboot Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_DMBOOT\0000
CRC Disk Filter Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_CRCDISK\0000
cpqcissm Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_CPQCISSM\0000
Beep Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_BEEP\0000
AFD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_AFD\0000
Generic volume Yes VOLUME 5.2.3790.1828 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{235BD4BB-B766-4D21-B59A-9EA4553F0AEB}
Generic volume Yes VOLUME 5.2.3790.1828 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{45D1C6B1-18CF-41A2-A8E0-EC3D60EC970E}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A44EC383-DE9A-4C97-8CC2-EDFDECABA0E7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CB061B95-C308-47D6-9BB1-ABD65D453C55}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3E92641B-1A5B-483D-91F4-8A6F39B96205}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{49656E59-8960-4B00-A563-A747877F5404}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D23CE65-586C-404E-95AE-94026131C75C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{81A15B2C-0FA6-42DC-BD30-7555C5D78375}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F956BEB5-FD84-4F97-9E60-CB9762233AD1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{37455DE0-065C-4F17-A7BC-1C0EF1412208}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7530A6D4-2130-418D-AF6E-62D95B958DFC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{157C9244-7657-43AA-83F9-1EEFD8FC48D8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A7632FB5-B8F0-49EA-9C8D-4426028E71F9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C43436DC-DEAD-45B1-9377-E10E376FBA11}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E17A0F7C-D2E8-45F8-B015-9B8DD65BAF1C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CF613193-52A2-4F94-AF16-A754F3DEFBA5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D82E3DE5-9B47-4291-98A5-444F696D9C3B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{64BB2810-1CE7-4658-A62C-CCD2C671CFDB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FCCD6008-D5F8-4CA6-9381-0182CFC6E879}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4B122E5D-B8E0-4CE5-B7F5-2E8EF23703C4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DA676FE3-185B-4657-9096-32878D27AE6A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{70DAD32F-6885-4CAB-98FC-B58149832942}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6685ADAA-F8C2-449F-8AC9-4B7F05EAB336}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0DC83CA2-5B88-4881-AF8D-8D81D21AA759}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CB6BE608-2CF3-4D4D-95A3-38626FEA4B13}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B3DA709B-0728-437D-BE7C-5F3DDF722459}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{100D8565-E119-449B-AF3F-4ED591B1DC4A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{720959DA-3CEE-4424-9D42-C19D607D6CA2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{852BD3F7-3A42-4808-9A17-B44B0CDC5442}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{235BE4C6-9A13-469D-AF62-CFC3A018DD16}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F721EE61-AC35-439F-9235-8EA6D5D07F41}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3D19D516-42B4-48B7-869F-C2CB87646ECE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D6294944-E655-4741-A45A-3A529A78E89D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2F6C7974-43D8-4905-8A51-5BA7CEB796D2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A9BFD441-771B-474E-B906-9850A9E72892}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F219AC34-7365-4C91-99C8-AFCBD1CE38F5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CD3F2D05-E35A-448A-9E44-1F757EC064C7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{33D0241B-7E3C-42A5-9C90-F7B5DA37CD3E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DBF94855-909D-44C5-864C-F87DE916E95D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{09E68240-0D94-4FC6-BFF6-618E7A7CA233}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{83CE9C08-4994-44EE-BCAB-AF0C5D50CACA}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A8E36925-952A-476F-A661-979609C19C95}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{65BA1681-0666-4F59-B7F0-7DA5CA08CDFD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D70C28F8-6324-414D-A519-F2BFBF91080E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{80490FE2-934B-444D-B1B4-4C963494F1FA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{85255E04-D5E7-4677-88E8-54CAD9BD0070}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A61F7CB5-86C4-467E-B75C-135627B1C99E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F98BAE84-B601-4490-AA29-168907D94123}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3F498D67-A2B9-4637-8FA0-4D59D882560D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{44BA5AC3-81F7-4C73-90FE-C9EB7AC7FDA7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2EF02DBC-EA30-4FC2-8D3C-464302B7C73B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A34AAF34-A12F-42C1-865D-18705C34793E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{95A9BF05-33B4-49AC-9025-17968395C841}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1ED97587-5CE5-4335-AE07-2E9FD7CEA1CD}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CFA503C2-C801-492C-B5FB-AB0023B12811}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9088A201-6C38-41F0-B545-A7F4D6C1C7CC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{02CF3A0D-3881-4525-9E7C-771855075BF5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A1270E2F-B76B-412B-9163-86F6BBA1D845}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{65D7CEE4-CF5C-4F2F-BB21-04D4221B902D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AF9BC0C3-5CC4-4CA4-828B-F7FF44B757BD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5D5E3FCD-6AF1-44FC-BD46-5339EA970EA6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1AF85B30-114E-4386-9030-57F4600D97C5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9BAD3042-8B07-496F-A85F-6698B0DE45F4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0C4AF375-7918-4DA3-A85D-72C9DE7BC91B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E81CF22C-ADAC-4A97-A8D4-65D9C65E691F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{09E1612D-3737-4581-9EB3-2A9723AD18D2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BC168DF9-B2AA-4D27-B1FC-53D5E7370C4D}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{116D1C46-4BFB-47EC-A0F4-2AF136584A53}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0209CF7F-76B0-4BA5-829F-603D48A38691}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0F2005AB-1F32-4081-A3D1-F742AE4765A8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3B8FD9FB-2CAE-4150-B7ED-56764F809473}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FB7309AD-5FB0-4F37-B827-C07FC0BABEB6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5E25FDAC-CF76-4244-BBA9-640F6F7ED486}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{140E30A9-0889-4204-90BC-ED626DF95297}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{118DEFDA-BC61-4E49-854A-1196C02915FC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{603B617D-9AC9-465F-8500-25205D3DE9CD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0BE307CA-CE87-408F-9E97-52A2C417D545}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AF623BD4-6222-491C-A6AE-3F0032F4A0C1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CEB81B76-8272-4FAE-9021-3C100E755C20}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C21F1FE7-6785-4045-9463-77D263E95031}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BD47C52E-BEF2-4C01-A481-D02B1F62E276}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4459554C-A8BE-4C25-AC3B-61A81C75CA9F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{92772CDA-B1CC-40B9-B9D0-C20A860117B4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E1F02CD3-8936-44EE-AF87-47A05AF3B28A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9271A603-ED14-4853-A815-1FAC50A69710}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9494C285-D304-4CA5-A791-2AA776168800}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BBDADD0E-6C1E-4EF9-A76B-0CCAE79B8444}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8CF3E463-7A00-4634-A720-08626FDB581F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AEBC38EC-3A4B-450D-A57E-3BD57944FA7A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{88F7B01D-243E-40C8-A8EB-B2925010709A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4CFBF315-047D-41A6-A0B2-53153954B980}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A55E3F3E-AE97-43AC-A196-DFEF5B1B67B8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9218A8B9-8F31-45D1-AFF0-042A6C769BD3}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0BEF5AD-8EF7-4704-850F-E94D858D751B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1065C764-9E3A-463B-895D-0BDD34F4A613}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3D5F0FBA-215A-4BA6-8C28-34AF6E2F518B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E48687BF-7D61-4479-BFCB-027BD7913D3F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{799214B7-9C89-48DB-B6B9-5162CCF8C770}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA23492A-25DD-4E04-9DF6-DFEA682C0433}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E555D766-DA87-49C4-840A-EB44139C7260}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B2AE4C1D-8608-45F6-807F-6CD8D69FE046}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B96C836D-3570-41BD-99C2-444C539A94D7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3A2B48CB-D0B7-432E-835D-34A51A2A9BCA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6201256E-96D6-4567-A18A-560E47D390EF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0636C323-DC18-4A7C-B08D-AACD8C0C205F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B8736B8A-D5F4-4A9E-B13A-4221120F3BFA}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{010248E0-5F81-4A80-9048-8DA9223FE07A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C6924BA0-A0BE-4732-92BB-827D26C82153}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{572A0C95-BC19-44D1-8D65-32986BF8E6DD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5604AAAE-67BF-4553-9B03-DE78A5536BD7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9561B17A-2CB7-4073-A1D4-1B3FB43B5C22}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{270B316B-9458-42E4-B980-D281EB247D3C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{63F10A17-F554-4DC5-BA09-3EC450CF6161}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3F8B4535-2350-4329-8BB7-9EC1E86528CF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{61109B84-60A7-46A4-A4D0-3C4575BF3250}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C0806ADF-B7EC-4BEE-A959-28862A3FAE39}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CF33131E-DB6C-4B7A-85D3-7D85F77A1C1E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CCFF3496-6E5F-4857-967D-8C4DBCB87C5D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{28980AE8-EA5D-43FB-B024-D4D13E439BF1}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BE2AE2D9-060E-4C46-9660-B8588E82B429}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{359E2D68-11DF-4B4D-8B30-BAE424BEDA48}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{82256D81-5C0C-49AB-97C9-A31A14D30DD1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C3C89440-E8A8-4FB4-AE44-697A7AE60674}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9E72F435-2D81-480E-8B40-AF7F77052EFA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A2B35BCC-BC36-4277-842A-F61B107FFFD6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6DC8F726-3CED-4402-8B80-A3B3CBF58A17}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4D9BF746-7C3B-40CB-AE04-A665991D7103}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{381336A9-56F0-4230-98C7-AD0E44AD3888}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{61FAD7A4-8091-4964-9EB5-4927353CCAFC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6D361BC0-8810-4FF6-B023-C1F64B936443}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{04EBB374-E683-4B08-B29D-F293616CDD82}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8A7C0EED-7665-4631-8DE0-1CF7B9E9A807}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7BC51164-05C6-4F1C-B482-8785B8BEA0C9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6FA8269D-5864-48A1-B00C-4826D554B0DE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7CC5229E-EAFE-481D-A33D-7544A9F64F70}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EC74B54A-8B09-4AFD-A9C0-83DB30C38EDB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B2838E7F-C592-42FE-B537-94B723949351}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8B6ACA85-B360-49AD-9B10-D2C17DDBAEDF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{732FE309-8825-421B-A09B-741AE9EEA1FE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C60A20A2-240A-4400-B697-710681E302CA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{24E24EA8-E7D1-46A6-A58C-266C60582B48}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E00B6B53-4D1F-4184-9F05-76BACA9B4C20}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3CAA6971-EAA7-4C92-9940-8960193EB765}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{783C34DE-3276-48CE-AFBD-5ADC8B24A048}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FC026CD6-35B5-4931-A57D-489A098B9F0A}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D136DC94-04FF-4D00-AC65-9A7F4E880527}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1FCAB4CE-B7BC-446A-9274-DAC5F2D7C6FF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7BCC4B39-A770-4C93-BE2C-A135A55B4AB5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{63B71A90-C10A-4C81-AE2A-0ACCBE0D79D5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0BA8023D-77E2-4B97-8C9D-5B7D5056040A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{39A7C770-C0F6-4422-B6D6-4B0320EF0DC0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{942F7F4C-211E-4937-94E0-132403098334}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4B1BD08-6043-4100-85C4-5E6B58FD3A39}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{94ADF6A0-C407-4F8D-A1D5-4084CB41D265}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{72063B6E-097D-4F67-9E05-354ADE734108}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4EFA4926-0DFE-452A-BD7B-C8C83B2642F5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D8CDAC09-E465-4453-B7C3-B916FF7F5354}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CF967653-A484-44DE-B5C6-D366618BCA17}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D7276126-35AD-4C10-A0FC-F24052C80F82}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2F2596C7-5499-4074-A9DA-810483B074F5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D6723549-3987-4D1D-8E67-97AAFE453E17}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9953EC68-5845-4934-AA2A-E6C43160550C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{293EC9E1-60F5-4D45-91E9-CC94B4481DA1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D24E96A-1D5F-40ED-8632-184E269C168B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{89CC9149-D5D9-4447-A054-808F221A243F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C36F1EFF-5F12-4EC6-82D5-606AF9BF4D42}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{84C06EAB-D70A-4BD7-BA99-59C7CEF86A7D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{78721D9E-326E-4689-9411-5B790DD9A052}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0344D5EF-3A16-429C-95D1-92E50A859D62}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{30380199-4DB2-4A99-8E40-8BEBF59B37A4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{99F16DA7-484A-4D1A-9AE3-C1D296FE15E4}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{96633AE6-C7CF-4DBA-99AE-14A758E46690}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C88CE3FD-6C1E-494C-B7EC-C28869C98777}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{72071F8F-EEE1-4772-A0E7-8A0B280E4BA4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8D640CDD-1F71-4F06-9745-B7BAE507C0B4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{41047961-1E1F-4043-975A-18FBB91B35F4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9CE846F4-2946-46C2-B97A-01B35202BE88}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C105D2D6-C1C7-4580-8319-F225C51CC284}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7F593814-7F65-46D7-8492-EAC60949CC19}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D6DF8CD0-BC15-4175-BCD1-3D6B4F3972F1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{134D44FF-62FD-430A-A9CF-8A615BA3D74B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D35095AE-3000-4BAA-94CC-1083DAD8732D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FD029669-AEB7-48A8-9EAE-C9742A4F10DC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4FDB40AB-78AA-4780-A0B3-0BF5EB089A71}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3940FB62-9F70-429D-90FF-B1D0B3F43AEA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{876D29D1-29DB-4E22-91BF-537934B20AD9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3F51F6CC-AACA-4E5F-9F2E-3EB532D5E005}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91DAF123-8E21-40DE-B543-71ECF0D77C6B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FE0F4940-1F00-4DCE-8749-27A6BA35639D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AE9BB84D-7203-49F4-BEEA-5A0A8289E8A2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9751CC11-AF19-41A2-AC1D-8DDD22065FF8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7F239308-20E7-4D55-B10A-6895D72F5D0F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B751046A-20DA-4616-8A5F-10795F434A88}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4EDDC5D6-4FBA-4EB8-AE82-6B4C02FE5004}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{760A8E3F-BB7F-4D99-9176-BF3C4FD49494}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{099458EF-FD18-41CA-8F54-FCFCAA4DB4ED}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6EE28355-B0B5-4B59-BEA1-17E84AA4A82C}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{928D3512
-FADC-49F0-AABE-E910BE491B37}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{40362580
-03FC-4024-8E3E-72CB3ED1A0C1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{48B7B602
-CB04-4EA5-891C-7620ECBCE68E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D79778B9
-788F-4B1E-8DC6-7667A9EDEC06}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9AAAAD1
D-AAD1-4265-B5E3-2C8B41194596}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2D26C740
-21A3-4A5A-ACF6-C7DE4B4F6BB7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9DF52862
-B136-4FA3-B30A-E5A80D84C464}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C32195A7
-7807-4A92-8D5C-B35B83722D43}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DFB495C2
-70A8-4ECA-ADBA-35A433A2B915}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B256555D
-9589-4A0A-AE72-F7AFC0E1A7B3}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0B8DD2E
5-C45F-480B-A1EE-A0ACA1E1F97B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EFE51424
-5B16-4EAA-BCEC-56C2D0EFF0EA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{305B084A
-B3A2-4FDB-9134-7CE86A98835A}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4C677034
-FC03-4300-878E-A77D84F7113C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FD85738D
-64B4-4B37-9E96-47DDC05E76F0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D527EF67
-38ED-41A3-8462-3A0B2914ED5F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{03AC9720
-3BBB-44F4-AB59-EC320EA19FCF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6D8703A5
-A17D-44D4-A92D-333E5DC7F4F0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{926729E8
-096D-4EE3-A0FE-B066ED087EFF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{25C15BDE
-B98F-4DF7-8970-0CFD732DE098}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{30DA8083
-0C90-4703-8890-F2FBAA98EDF6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F57B9F7B
-4CF7-4394-820C-AF0CDAAE59B9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06EC3319
-2964-412F-BC2F-14C35E0EED5F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CF7F54C3
-1903-4CC9-B366-3AF9866736D8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CC81D602
-E48A-4CA1-A796-AACD4B414984}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1AFE55FB
-F05E-45CD-8115-0C5AB2DA4FB9}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{40092DF2
-80BC-47D8-8C3D-42D5110754C9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0EE6D95A
-60C0-4A0B-83DD-525110C50194}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{25401776
-F582-46BE-83C1-7596606E6669}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D1D90E2
C-E57D-41EE-92F3-BB3C1077FC2F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BF52EFA2
-375E-4B0C-9A89-529F33A362FC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{60A4952E
-D096-41DB-879F-8D88DBDDD7B8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5D725A0
D-739E-459C-A58D-F52F28A5B733}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6DB1C6FB
-D0B9-4E19-BAE1-BA8704855D97}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{122256C2
-DA55-485E-89B6-3F6E0266019C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{86FCEC80
-947F-4DCD-87C2-213A196F9475}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7E3DA480
-C876-4CAE-8E8A-D4707475EDBC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9FC19799
-73D8-4587-A019-27702480EE05}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{003C015B
-398F-47F1-B2C0-8E5530F2006D}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8FD247B7-617A-4973-9DB6-0A65B63566E5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BCA0C65C-C4D0-42C2-8079-AAC889105EEC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EF1B58A8-5530-446D-8516-9ADB39542CF0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C58C856F-D2BA-4123-8368-07FC18096B98}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{18FD3885-2A24-492D-B1EF-17B4B2BB72AC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{96C22AFB-EEDB-4451-842C-0988A3A91F4A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{14331FD3-6696-40B7-AA6A-47B8AD030BF1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C36E0007-AF0A-40CC-95C2-58D850759883}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{760835F7-AE5E-4734-AF6E-A273FE5D69D0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E290DB0D-6C9E-43C1-BBCD-FE87640F3EA2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{01E1AF32-8F86-4B8B-8C4E-95C278C5A531}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D316D4E8-62C8-4471-839B-F759102AFF7B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3CF31B10-4935-4D0A-9745-EF1991BC4993}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{506043D1-48C4-4EA1-8271-C377BD6C5762}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{99AB9407-AFEB-4F4D-AC6C-C5CC1B4C9FB8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F1CA8DE2-A9F4-4994-AB9A-262FA6A58639}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4BB1A0B0-648B-47DD-BBBD-D490F3C48453}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3B2C46AF-62CC-403C-B612-735210C90033}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CCA08D44-CB0F-4BC5-8AE1-2A0ED04338AF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C92C2C7F-C885-4F7A-96FD-5A02D9685B2D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CC46D95E-B0A3-407E-91DA-3A15B9696F5C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C34B3E64-D7AB-435C-9EFB-1F4063D698D2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4981D0C9-C74E-423F-8128-4930126A9B5F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3172CDD1-AEDF-4AA9-BDBB-F04ABD5CCFDE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C33CB335-AF67-41DA-88C6-A5998A377625}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2ED8FF8C-F167-4BBD-B33B-8DAB0FEC9ED4}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F5C7280A-704B-4C0E-85EC-5B3D465775 78E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{32B9EAE7-FCEA-4DDC-A2E0-4E8216EE408D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6A96E57F-0217-4EB5-9474-04A77AAE43BE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C6B24F92-D0E7-4927-91F7-91418AC8F3C2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DF863D62-8A2E-4B2E-A634-1A7545075627}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BB703BE3-4ECD-46D8-9FA0-635A864E4511}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E3C5C6D0-0A4B-4D5B-A9E2-FB32A7173850}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7AF96630-A6E1-4EBF-8566-4587A3C93E67}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AF2FA97E-C547-4B04-B37E-E02EB2609BAC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FABA8958-07F6-479A-A0F5-6DD5B4236044}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F4D056D6-1AC4-4007-92F7-B9501AC5002E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{43F58249-6FF0-4AE2-ADB9-DD78AA966EDC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{58EA2FB2-8D8E-4332-932D-C3826C5FA09C}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B8B6F424-70CD-46F1-AC87-73805CE7AB66}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6F56956E-3C10-4A95-8862-249E39B5EA43}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C4E3B394-F056-4F91-8E7A-C9A78F02F469}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{80CD898B-5860-4BF9-AE29-0738789D1CCB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{71BE3CCF-8FEC-43CB-BF17-AB7045DC171A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4AC62E3F-9A43-46B0-B8A1-E6D9F080765F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{347A4254-08DF-4F45-9A91-EFB9670623DA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{131A4AA5-BB3A-42F4-93CF-0CA3CEE28210}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D289D8DC-21E5-4230-A172-D30ADE33FD04}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{33A4FD4F-B97B-4353-AC1B-2281A0DD60E2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3DB45359-81E3-4976-8276-D3264C0117F0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B9D5347D-1CDC-4214-9214-24C1017AA682}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{38A6DDE0-F61A-4648-B4BC-60B1A3138D6E}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DBB47209-011C-44BE-BEA2-44EBA37FA3BE}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{00E66B9A-5302-45FB-B4FD-19F4E7621EB6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7D66BAAF-58FF-4BCD-80DD-E6F2DFAF08C1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC89C41F-2075-441B-8CCC-0F39D2548716}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8C940522-6521-48B9-8124-1441D8BEFDF2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9FB4B2D6-1798-4291-AA53-5607CC206E85}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{61A8B774-E5A2-4DC1-83C8-1F5A0695CF07}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0437D990-7F85-4109-8428-BB134F97BFA7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0AFD7546-0ACF-4EA6-9D74-ABAD1509A8B2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6FBF4D89-D140-4321-AAEE-2F089621AAB9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DFA79945-B1E9-4373-A38B-B75EC224A98F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{75BFF5A5-69F1-4793-9F97-9DCB74F6DF23}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5121AF22-9CC5-4B30-830D-D52AC354A8B1}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2AD9F330-109D-4CD9-A8E1-66FCCD073BB3}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06F316AA-F176-420C-AA4A-3783A2DE4A7A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{48511D97-FB52-47FC-AE25-879B1C0C733F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{33383701-40F3-4E09-A611-116F5BD66303}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2E11F995-21CF-4FB1-864B-CAECEE6579C9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EA085F8C-0ADE-4454-948B-5BF7340D8B07}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{18FCCD4D-76B9-410C-A89D-32F421BA6158}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3EF33453-D870-48AC-96BD-369272DB53CA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B8AE4828-EF66-41A3-8107-1B3D8570C541}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA4EE8F8-6F58-4D44-A578-C045D21A3E7E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06D6FE3F-32C5-4C2F-9A3A-8D4D4C900E2E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3BE43CB9-ABB9-4F71-83FB-576560A03402}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DEE58A6D-93E2-41B6-ACC9-237AF332596B}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{66B0CCFB-A105-41E9-BD89-3CFC098CC2B9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{843D0892-8DD2-4C86-91CD-FE942D603A51}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E45A93D3-EAB7-4A1E-8A15-16DE389F311A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CD623994-583E-4E5F-99CE-257B95BC3948}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F7777657-EDAC-4182-BE35-5CFC087CEBDC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{72C5B367-8460-43E7-A5E9-62507511F002}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{75B6BE9B-DC0C-40A6-8336-E7DA46319A56}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2A2ECEB9-0539-4783-A838-017024C0700F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8C49F6FA-B090-4AE2-89FE-62B274399165}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{09D423D1-22F4-45E5-8B93-E49BACBD7AB5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{17CBAF95-15A7-4E39-883A-EB663F8403C0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7C94304B-39C5-426C-95CF-47418A520EA5}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{764B756C-F409-4089-9189-AD46EBAF1E5D}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D599CD6
Generic volume Yes VOLUME 5.2.3790.1828
8-1051-4B16-9B1C-8EBB841EDE90}
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B3388607-0706-4EB1-A859-BA95464E3256}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EA194AB5-C5A4-4B11-BCCD-B1A6FB146903}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A563CBFA-FF5D-45E6-B014-75DA8D1DF2DC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{09EE812E-78E1-4935-BC0B-68FA63C4E477}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2FB8EE2E-EA20-4ADE-9C7F-78D47D4ACEAF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D992C54A-AD32-436A-B6F6-59517FE64D29}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{834642ED-0E20-40C6-9CA5-B2827041E295}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B2D40731-B774-49B5-9AB2-31FFE8E9E69A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3FBB84FC-1B01-4A63-B01F-0486F37E940D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3B83DC8A-01E5-4976-9B70-B254D9B1F28A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9F2B447F-B83D-4985-969D-49B0CB96A2C7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8F5F8407-0C10-4829-AB42-81DF2B927D3B}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4C2426C4
Generic volume Yes VOLUME 5.2.3790.1828
-5EBF-41DF-849F-2F0C4A9664}
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{722ED8C7-41D5-4C77-81F2-90F7170B50B3}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A31A3742-B1BD-4161-83F4-2379ED031BE7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BFAFFFD1-5DE3-43EC-AEAD-A225519171D8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5A9DC812-C0D1-431D-8450-2541F73DBED1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1739E1B2-C713-481E-955B-6CE6255A71D4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D9DB1E5-55FB-401C-B1A0-BB452C80826F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{68F88A21-CD79-4034-8561-BC296D1AA67F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3BE624B2-5174-472A-BF7C-49734BEF76E0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{094FA683-2467-46EC-8606-B350B140D8DD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{614F1C38-D36A-44A1-844D-B98EACD425EC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{81F39E6C-2876-40CA-A06E-70F14CA24383}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BDEF0C32-D821-40F8-87D6-64E4DFD2BF34}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{74C3C6C0-432E-4218-B970-3D4DF9653DDF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C5B5F77B-320F-4468-B1CB-E3D44FB3AB43}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A95AD0FC-C91D-49F9-8D56-F0DC56F98831}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ED0453F2-EB73-4E15-9B5D-71FFA0445035}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{26926D8B-D774-4BB2-8ACC-572076907D13}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{730D1E68-9FE3-43D7-B9B1-D80E0A8BDC01}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3E081287-10A2-4F27-BF89-23BD8B49A772}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C4AC5ACE-9383-46A2-80A5-F332CD5F5436}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DDAFAC10-41A7-40B7-867E-8C377632E4B2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F6835740-5397-4460-B55C-5D7535FFF99E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A7ED69AF-294D-4702-AF9E-83B32191F1B0}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{412B3090-4B2D-4D02-8832-0A7F435B1BB1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{983CA2CA-BBAA-4225-A4E9-12769477E429}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5FE55F19-CE56-4944-A56C-E2127A0F6A77}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C6CBBAF5-EBEE-47DF-9189-DE82A50DB989}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5808EE42-E1C7-4169-9581-947A5FCA9E67}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D08DA0C9-4A2B-412F-BDF6-5D249B24D069}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CA086117-9278-40BE-A071-1EEDA3BB5ED4}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{493C6ACC-A2EB-49D0-9CA5-31D27814C883}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2A57B12E-2266-4B68-8B77-79ADF1FF06EC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A74DBF5C-F179-42E4-B4DC-91AF11D8902F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{32FD8808-37F6-4D3B-ABFA-381F23CC6F61}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F51E0E77-D8B5-431B-B32D-2AE5D58725B2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{58D746B2-21B2-4120-A890-BAF8F419D123}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{47F6B2DB-47A9-4BBE-8B87-9A4D9A5DFBB8}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{09EE1407-3A38-4C1D-8E4C-2EC5796252DA}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BEB9C98B-0912-452E-BB18-858B46B3663A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C5370EC2-5380-4E68-92FE-3F15D4D9F158}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B50F8FE5-25FD-4B36-916F-92D8E4B8B441}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0F24626-7BE3-4146-BC3B-FDFAC35FA7AA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3E0FED01-86EE-471B-9F08-39A3324B0BB7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FE553A2C-9ED7-45AD-A8F2-9CBFFFE5C001}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{16ECE5D6-28F4-4013-92AD-085AADEDBAF9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C0984D57-6D5A-4CEE-A81A-64E526A3905C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FB4EB603-2684-4013-A19B-E8E426B3D1E3}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F9E80C40-F335-4E3A-B588-C0C86F81803D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C4EBDD56-6FDD-4460-9D52-52BFFF3C5AD2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AE30AB06-BD89-4195-AFBD-AC6B4F8549D6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F06DE4C8-E40A-4F23-A5DB-D6A0DC24E447}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{008AFB70-BA6A-442B-B2FD-C1534CC2B0BB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EC430F3A-74F7-491C-9CF3-5E15F04722EA}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CAA2B047-0AFB-43AB-9AE6-00B6DED87D77}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{18A1D02C-9477-42FA-9742-F2FB31D7FEF2}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{25E99D6A-6283-4AD1-9ECC-BE2304C34DBF}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9DDC5CB8-C3B4-4A38-B780-9E4FE237B401}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C8D6C28D-C9BB-4B6F-BC9A-433C7E477EFD}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C00187C5-516C-41D8-8D39-EFB9FF748434}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5474E4B0-6CC0-43B9-82EC-84EB66561B1B}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{914E3F4A-E28A-4056-BC6B-0BEB39D807F6}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E9949C59-FCE9-4D9C-9150-2961611F34D9}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8C999EB8-3156-4F52-90DB-415E6E1B3672}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8FA36A56-48B9-48CA-A11F-555A33EEFE13}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C3287728-B399-4C42-A45E-6A2F2DE0976A}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9B2EE41D-BE9B-42A1-A332-81BA4B6AF017}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A702A7FD-B993-4772-8BF0-DA59B103A1C1}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F8D74A20-0F95-466F-96DF-4D021107C3CB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA64997B-9B74-4EBC-9EE2-D7FA8F52F41D}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FF337C30-B781-439C-9AD5-06D28203A42C}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0783F8A3-E7AA-4614-9C36-5F989B32D0DB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A4276743-72C8-4C9F-A4B3-69562DC99C81}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{16296498-51CC-44B8-9AB4-ED19A1733F80}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9677BC95-934E-4884-88E1-FF1FA2D95B3F}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E7C82D6A-D1CD-4A70-8DB4-390DCB775854}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7B648FBA-DB30-4F76-AB2E-4DA99A03CF31}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F2FBA3EC-3B8D-460A-9CFC-E7B9BAA7BA99}

Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FF55362B-6299-4382-ACD-4782A28B4240}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{66CDE3A6-E2DA-4A7D-925C-AF1920AFA9FB}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B29D17AA-35AE-4332-9C68-BF9105FF14D7}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ABBC47EA-9A41-47F1-8DD1-2B9D5EBCC11E}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0B359E51-7222-4089-96C3-FA754813A239}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE655DF5D4OFFSET7E00LENGTH87A669800
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{71EB7360-F347-01C4-507B-9E5F8078F531}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A3D01CE0-F343-01C4-F1B3-12714F758821}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6A45BF80-F347-01C4-92E0-3C772E43AC40}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CC4C69C4-B58D-11D8-BD2A-000000000000}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B3DC4904-B58D-11D8-BD2A-000000000000}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B3DC4260-B58D-11D8-BD2A-000000000000}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B343C8C8-B58D-11D8-BD2A-000000000000}

Volume Manager Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ROOT\FTDISK\0000
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#{3A199798-85F5-
4313-9DBF-A0FAC20FE6BC}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#{AAA99A71-9232-
47A4-AAED-400EE4014001}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#{E1255DE0-A571-
4AB8-B174-3229D9D17390}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#{7FDBC91F-7897-
4B99-B452-D39FABCBF840}
Generic volume Yes VOLUME 5.2.3790.1828
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#{FF345629-2466-
47AD-9975-EA6D6FAA6EB8}
Logical Disk Manager Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ROOT\DMIO\0000
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\63

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\62

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\61

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\60

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&318659E
F&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&318659E
F&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&199
05E4A&0&2008
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&122A5B
DB&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&122A5B
DB&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&199
05E4A&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&275
320C0&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\FD
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&362715
A2&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&362715
A2&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&FCF
5F08&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4CD955
08&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4CD955
08&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&FCF
5F08&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&23E
568A8&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\FA
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\F7
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\F4
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&1450FCFF&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&394
9F8FA&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\F1
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&BD186B5&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&369
0080C&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\EE
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\E01
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&124050F
E&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&124050FE&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13DF1870&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&32D9652A&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&32D9652A&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13DF1870&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&19D09E3&0&08
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\EB
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&15575BF4&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&15575BF48&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1DDED019&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2E5959D6&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2E5959D6&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1DDED019&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&78745A4&0&08
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E8
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E6
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E4
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E2
PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\E0
Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\E00
Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0005\E
Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\59

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\58

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\57

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\56
Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\55

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\54

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\53

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\52

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A493398&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A493398&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&214639D8&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6DA58E88&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6DA58E88&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&214639D8&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&336
50587&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\DD
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3456336
D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3456336
D&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&348
D8DDD&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&10C382
BB&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&10C382
BB&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&348
D8DDD&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&36D
2BD9F&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\DA
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\D7
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\D4
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&32E4DACF&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&37
B34F&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\D1
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&290BCADC&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&331
D07A6&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\CE
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\C01
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1531E4
AF&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1531E4
AF&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&301
D439F&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2FE7D1
79&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2FE7D1
79&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&301
D439F&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&2FA
F4F8E&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\CB
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6FFC99
A&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6FFC99
A&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&273
093EA&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1162229
&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1162229
&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&273
093EA&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&268
B0007&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\C8

PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\C6
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\C4
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\C2
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\C0
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\C00
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0005\C
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\51

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\50

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\49

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\48

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\47

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\46

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\45

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\44

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2961EE4
&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2961EE4
&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&36
DED50F&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6E8CD3
D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&6E8CD3
D&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&36D
ED50F&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&552
BFDC&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\BD
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2C9208
23&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2C9208
23&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&F78
FC9&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&E62A2C
E&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&E62A2C
F&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&F78
FC9&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&32D
A5213&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\BA
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\B7
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\B4
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&39F8735B&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&22D
A925C&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\B1
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&55F4908&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&279
B1EA1&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\AE

Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\A01

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&94B57D
3&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&94B57D
3&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&466
FDC1&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4C9897
5&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4C9897
5&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&466
FDC1&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&2B0
8D6B9&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\AB
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&8926DD
0&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&8926DD
0&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&FD9
6073&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&5827378
&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&5827378
&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&FD9
6073&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&342
D2640&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\A8
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\A6
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_4.3
2\68&229DFB7B&0&001
StorageWorks MSA1000 Yes SYSTEM 5.30.0.64
4/12/2005 Hewlett-Packard Company oem19.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.32\6&22
9DFB7B&0&000
Emulex LightPulse HBA - Storport Miniport Driver Yes
SCSIADAPTER 6.1.10.4 3/21/2005 Emulex
oem18.inf Not Available
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&24E
12DB3&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\A4
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\A2

PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\A0
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\A00
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0005\A
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\43

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\42

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\41

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\40

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\39

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\38

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\37

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\36

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A66DF
E3&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A66DF
E3&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1EB0D7F0&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1AB2D645&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1AB2D645&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1EB0D7F0&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&1A9D3CBE&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\9D

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&20FA00C7&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&20FA00C7&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13220302&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&241FB561&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&241FB5618&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13220302&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&36001D45&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\9A

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\97

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\94

Qlogic processor device Yes SYSTEM 5.2.3790.1828 10/1/2002 QLOGIC scsidev.inf Not Available SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\6&18F9A600&0&07F0

QLogic QLA23xx PCI Fibre Channel Adapter Yes SCSIADAPTER 8.2.2.10 4/16/2003 QLogic oem3.inf Not Available PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&A6512F5&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\91

Qlogic processor device Yes SYSTEM 5.2.3790.1828 10/1/2002 QLOGIC scsidev.inf Not Available SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\6&372CE62D&0&07F0

QLogic QLA23xx PCI Fibre Channel Adapter Yes SCSIADAPTER 8.2.2.10 4/16/2003 QLogic oem3.inf Not Available PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&7AB22207&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\8E

Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\801

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2726FC348&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2726FC348&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&14635F0&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&23A70598&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&23A70598&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&14635F0&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&3081EFE8&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\8B

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3634E8C0&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3634E8C0&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&20DD8E1E&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&EE4CD68&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&EE4CD68&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&20DD8E1E&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&143D29A6&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\88

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\86

Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard disk drives) disk.inf Not Available SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_4.32\6&2C4C6E09&0&001

StorageWorks MSA1000 Yes SYSTEM 5.30.0.64 4/12/2005 Hewlett-Packard Company oem19.inf Not Available SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.32\6&2C4C6E09&0&000

Emulex LightPulse HBA - Storport Miniport Driver Yes SCSIADAPTER 6.1.10.4 3/21/2005 Emulex oem18.inf Not Available PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&36A998EE&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\84

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\82

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\80

Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\800

Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0005\8

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\35

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\34

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\33

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\32

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\31

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\30

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\29

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\28

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&397488D1&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&397488D1&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1B9584B9&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&BA52D578&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&BA52D578&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1B9584B9&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&4801F82&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\7D

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2EFCA026&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2EFCA026&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&26156FBC&0&2808

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&161D1602&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&161D1602&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&26156FBC&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&7ED
D79A&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\7A
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\77
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\74
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&FFB230A&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&23A
D32B6&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\71
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&1856D29B&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&266
723A4&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\6E
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\601
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&193C89
83&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&193C89
83&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&33C
2A78D&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A742C
47&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A742C
47&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&33C
2A78D&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&2A3
6365F&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\6B
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A7983
6D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2A7983
6D&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3B9
40DB4&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1937325
D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1937325
D&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3B9
40DB4&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&19D
51C05&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\68
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\66
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\64
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\62
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\60
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\600
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0005\6
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\27

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\26

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\25

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\24

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\23

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\22

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\21

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\20

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1295BE
A6&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1295BE
A6&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&27E
BA306&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3283F78
2&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&3283F78
2&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&27E
BA306&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&299
22629&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\5D

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&293CF9F
18&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&293CF9F
18&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&285
861AA&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1A73BB
D9&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1A73BB
D9&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&285
861AA&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&262
46E11&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\5A
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\57
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\54
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&1A490C64&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&370
AF391&0&08

PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\51
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&1CFE3B12&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&345
102A3&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\4E
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\401
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&18EF8EF
D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&18EF8EF
D&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3B7
BF75A&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2AC126
CD&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2AC126
CD&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3B7
BF75A&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&3DC
0F4C&0&08

PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\4B
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&35458B2
1&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&35458B2
1&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&6AA
0D5F&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&9D5358
2&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&9D5358
2&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&6AA
0D5F&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&548
403B&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\48
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\46
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\44
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\42

PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\40
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\400
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0005\4
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\19

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\18

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\17

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\16

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\15

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\14

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\13

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\12

Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&357E8C
51&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&357E8C
51&0&0000004000000000

Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&8A8
E882&0&2008
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&31FE95B
5&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&31FE95B
5&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&8A8
E882&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&312
6001E&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\3D
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&172E986
88&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&172E986
88&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&160
0BE56&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2DEB1D
C0&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&2DEB1D
C0&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1600BE56&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&3493B836&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\3A
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\37
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\34
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\6&1F0F1F00&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&8F8ADE6&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\31
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\6&29111EDF&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&355C0D0F&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\2E
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\201
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&28CBD6FB&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&28CBD6FB&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&16EA0F70&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1C4DDF2D&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1C4DDF2D&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&16EA0F70&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&31EE54F7&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\2B
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&33F8103B&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&33F8103B&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3533F53F&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1121A5ED&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1121A5ED&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3533F53F&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000008&REV_02\5&28CA0570&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\28
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\26
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_4.32\6&2B6D83F1&0&001
StorageWorks MSA1000 Yes SYSTEM 5.30.0.64
4/12/2005 Hewlett-Packard Company oem19.inf Not Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.32\6&2B6D83F1&0&000
Emulex LightPulse HBA - Storport Miniport Driver Yes
SCSIADAPTER 6.1.10.4 3/21/2005 Emulex oem18.inf Not Available
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&3815FDFD&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\24
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\22
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\20
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\200
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0005\2
Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\11

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2\10

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_9

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_8

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_7

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_6

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_5

Processor Yes PROCESSOR 5.2.3790.1828
10/1/2002 (Standard processor types)cpu.inf Not
Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_4

ACPI Fixed Feature Button Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\FIXEDBUTTON\2&DABA3FF&0
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&C1BAA7
3&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&C1BAA7
3&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&2B1
86594&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&38FE0B
B5&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&38FE0B
B5&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&2B1
86594&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&313
BA73&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\1D
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1DE528
B4&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1DE528
B4&0&0000004000000000
Smart Array 6400 Controller U320 Expansion Module (Non-Miniport)
No SCSIADAPTER 5.6.59.64 4/8/2003
Hewlett-Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&145
7B150&0&2808
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&27348D
74&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE 5.6.56.64
4/8/2003 Hewlett-Packard oem2.inf Not
Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&27348D
74&0&0000004000000000
Smart Array 6400 Controller (Non-Miniport) No
SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-
Packard oem1.inf Not Available
PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&145
7B150&0&2008
PCI standard PCI-to-PCI bridge Yes SYSTEM
5.2.3790.1828 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&351
9577C&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\1A
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\17
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\14
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&B550215&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&251
997C5&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\11
Qlogic processor device Yes SYSTEM 5.2.3790.1828
10/1/2002 QLOGIC scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&RE
V_\6&19BEC9F2&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes
SCSIADAPTER 8.2.2.10 4/16/2003 QLogic
oem3.inf Not Available
PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&363
3CCA9&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\E
Generic Bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0004\1
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_COMPAQ&PROD_MSA1000_VOLUME&REV_4.3
2\6&3E5D2D3&0&001
StorageWorks MSA1000 Yes SYSTEM 5.30.0.64
4/12/2005 Hewlett-Packard Company oem19.inf Not
Available
SCSI\ARRAY&VEN_COMPAQ&PROD_MSA1000&REV_4.32\6&3E
5D2D3&0&000
Emulex LightPulse HBA - Storport Miniport Driver Yes
SCSIADAPTER 6.1.10.4 3/21/2005 Emulex
oem18.inf Not Available
PCI\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\5&32
C61491&0&08
PCI bus Yes SYSTEM 5.2.3790.1828
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\HWP0002\B

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&AD4AAFD&0&0200004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&AD4AAFD&0&0100004000000000

Smart Array Logical Volume No DISKDRIVE 5.6.56.64 4/8/2003 Hewlett-Packard oem2.inf Not Available HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&AD4AAFD&0&0000004000000000

Smart Array 6400 Controller (Non-Miniport) No SCSIADAPTER 5.6.59.64 4/8/2003 Hewlett-Packard oem1.inf Not Available PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&266093A2&0&2008

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_1014&DEV_01A7&SUBSYS_00000000&REV_02\5&29A1C50A&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\8

LSI Logic 53C1010-66 Device Yes SCSIADAPTER 5.2.3790.1828 10/1/2002 LSI Logic Inc. pnpscsi.inf Not Available PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\5&137A5C4D&0&09

LSI Logic 53C1010-66 Device Yes SCSIADAPTER 5.2.3790.1828 10/1/2002 LSI Logic Inc. pnpscsi.inf Not Available PCI\VEN_1000&DEV_0021&SUBSYS_1340103C&REV_01\5&137A5C4D&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\6

Broadcom NetXtreme Gigabit Ethernet Yes NET 2.91.0.0 10/1/2002 Broadcom netb57xp.inf Not Available PCI\VEN_14E4&DEV_1645&SUBSYS_128A103C&REV_15\5&38EDBD97&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\4

Intel(R) PRO/1000 T Server Adapter Yes NET 6.3.6.31 10/1/2002 Intel nete1000.inf Not Available PCI\VEN_8086&DEV_1004&SUBSYS_10048086&REV_02\5&2A694D11&0&08

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\2

HP Baseboard Management Controller Interface Driver Yes SYSTEM 7.1.3790.0 3/18/2004 Hewlett Packard Co. oem12.inf Not Available ACPI\IPI0001\0

HP Core-Io Device 01 Yes SYSTEM 5.0.0.2 3/12/2003 HP oem8.inf Not Available PCI\VEN_1011&DEV_0019&SUBSYS_104F103C&REV_41\5&4F5EBC7&0&08

HP Management Processor Yes SYSTEM 5.0.3790.0 7/10/2003 Hewlett-Packard Co oem10.inf Not Available PCI\VEN_103C&DEV_1048&SUBSYS_1223103C&REV_02\5&4F5EBC7&0&00

PCI bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0002\0

Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0004\0

Generic Bus Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\HWP0005\0

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_3

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_2

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_1

Processor Yes PROCESSOR 5.2.3790.1828 10/1/2002 (Standard processor types)cpu.inf Not Available ACPI\GENUINEINTEL_-_IA64_FAMILY_31_MODEL_2_0

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661421

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620932

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662631

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620636

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04538867

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662816

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620644

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620931

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639122

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627405

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627480

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662945

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627469

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627429

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662567

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661520

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639154

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661533

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661336

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639045

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662815

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661341

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620770

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620709

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620585

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662700

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662825

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627576

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620643

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627572

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662570

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620635

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627436

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620702

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662841

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620833

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639130

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639159

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620821

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661359

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04639005

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661512

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661471

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627511

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620738

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04739725

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662929

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620957

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662609

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661487

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661477

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662810

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627462

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04587395

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661423

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662926

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662881

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662630

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627455

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04627456

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04620989

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661350

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04662909

Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available ACPI\HWP1000\A56E04661474

HP CSR OpRegion Driver No SYSTEM 1.1.0.0 6/5/2003 Hewlett-Packard oem11.inf Not Available ACPI\HWP5003\2&DABA3FF&0
ACPI Power Button Yes SYSTEM 5.2.3790.1828 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0C0C\2&DABA3FF&0
Microsoft ACPI-Compliant System Yes SYSTEM 5.2.3790.0 10/1/2002 Microsoft acpi.inf Not Available ACPI_HAL\PNP0C08\0
ACPI IA64-based PC Yes COMPUTER 5.2.3790.1828 10/1/2002 (Standard computers) hal.inf Not Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not Available Not Available Not Available Not Available Not Available HTREE\ROOT\0

[Environment Variables]

Variable Value User Name
ClusterLog C:\WIN1828\Cluster\cluster.log <SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
NUMBER_OF_PROCESSORS 64 <SYSTEM>
OS Windows_NT <SYSTEM>

Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\80\Tools\binn\;G:\Benchmarks\PerfTools\;G:\Benchmarks\SQLTools\;G:\Benchmarks\SystemSetUpTools\;C:\Program Files\Microsoft SQL Server\90\DTS\Binn\;C:\Program Files\Microsoft SQL Server\90\Tools\binn\;C:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\ <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE IA64 <SYSTEM>
PROCESSOR_IDENTIFIER ia64 Family 31 Model 2 Stepping 1, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 31 <SYSTEM>
PROCESSOR_REVISION 0201 <SYSTEM>
PROMOTIONRATION 1000000 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
lib C:\Program Files\SQLXML 4.0\bin\ <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE
Path C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn SQLORCA\Administrator
TEMP %USERPROFILE%\Local Settings\Temp SQLORCA\Administrator
TMP %USERPROFILE%\Local Settings\Temp SQLORCA\Administrator

[Print Jobs]

Document	Size	Owner	Notify	Status	Time Submitted	Start Time	Until Time	Elapsed Time	Pages Printed	Job ID	Priority	Parameters	Driver	Print Processor	Host Print Queue	Data Type	Name

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
O:	\\orcamaster\d$	Disk	Persistent Connection	
S:	\\prototype\d$ SQLORCA\Administrator	Disk	Persistent Connection	

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time	Version	Size	File Date
system idle process	Not Available	0	0	Not Available	Not Available	Not Available	Not Available	Not Available	Not Available
system	Not Available	4	8	0	2826240	Not Available	Not Available	Not Available	Not Available
smss.exe	Not Available	884	11	409600	2826240	11/21/2005 2:32 PM	Not Available	Not Available	Not Available
csrss.exe	Not Available	408	13	Not Available	Not Available	11/21/2005 2:34 PM	Not Available	Not Available	Not Available
winlogon.exe	c:\win1828\system32\winlogon.exe	472	13	409600	2826240	11/21/2005 2:34 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	664.50 KB (680,448 bytes)	3/30/2005 3:21 PM
services.exe	c:\win1828\system32\services.exe	664	9	409600	2826240	11/21/2005 2:34 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	300.00 KB (307,200 bytes)	3/30/2005 3:21 PM
lsass.exe	c:\win1828\system32\lsass.exe	708	9	409600	2826240	11/21/2005 2:34 PM	5.2.3790.0 (srv03_rtm.030324-2048)	15.00 KB (15,360 bytes)	3/30/2005 3:20 PM
svchost.exe	c:\win1828\system32\svchost.exe	1000	8	409600	2826240	11/21/2005 2:34 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	36.00 KB (36,864 bytes)	3/30/2005 3:21 PM
svchost.exe	Not Available	1136	8	Not Available	Not Available	11/21/2005 2:34 PM	Not Available	Not Available	Not Available
svchost.exe	Not Available	1208	8	Not Available	Not Available	11/21/2005 2:34 PM	Not Available	Not Available	Not Available
svchost.exe	Not Available	1232	8	Not Available	Not Available	11/21/2005 2:34 PM	Not Available	Not Available	Not Available
svchost.exe	c:\win1828\system32\svchost.exe	1380	8	409600	2826240	11/21/2005 2:34 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	36.00 KB (36,864 bytes)	3/30/2005 3:21 PM
msdtc.exe	Not Available	140	8	Not Available	Not Available	11/21/2005 2:35 PM	Not Available	Not Available	Not Available
svchost.exe	c:\win1828\system32\svchost.exe	432	8	409600	2826240	11/21/2005 2:35 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	36.00 KB (36,864 bytes)	3/30/2005 3:21 PM
svchost.exe	c:\win1828\system32\svchost.exe	296	8	409600	2826240	11/21/2005 2:35 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	36.00 KB (36,864 bytes)	3/30/2005 3:21 PM
alg.exe	Not Available	1948	8	Not Available	Not Available	11/21/2005 2:35 PM	Not Available	Not Available	Not Available
wmiprvse.exe	Not Available	1980	8	Not Available	Not Available	11/21/2005 2:35 PM	Not Available	Not Available	Not Available
sacsess.exe	c:\win1828\system32\sacsess.exe	2704	8	409600	2826240	11/21/2005 2:36 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	84.00 KB (86,016 bytes)	3/30/2005 3:21 PM
csrss.exe	Not Available	3888	13	Not Available	Not Available	11/21/2005 2:38 PM	Not Available	Not Available	Not Available
winlogon.exe	c:\win1828\system32\winlogon.exe	3912	13	409600	2826240	11/21/2005 2:38 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	664.50 KB (680,448 bytes)	3/30/2005 3:21 PM
rdpclip.exe	c:\win1828\system32\rdpclip.exe	1800	8	409600	2826240	11/21/2005 2:38 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	211.00 KB (216,064 bytes)	3/30/2005 5:32 PM
explorer.exe	c:\win1828\explorer.exe	2308	8	409600	2826240	11/21/2005 2:38 PM	6.00.3790.1828 (srv03_sp1_rtm.050321-2020)	1.64 MB (1,720,320 bytes)	3/30/2005 3:20 PM
mmc.exe	c:\win1828\system32\mmc.exe	2472	8	409600	2826240	11/21/2005 2:40 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	2.51 MB (2,631,168 bytes)	3/30/2005 3:20 PM
logon.scr	Not Available	836	4	Not Available	Not Available	11/21/2005 2:48 PM	Not Available	Not Available	Not Available
lputilnt.exe	c:\program files (x86)\emulex\lputilnt.exe	2408	8	409600	2826240	11/21/2005 2:59 PM	1,8,a,11	472.00 KB (483,328 bytes)	4/14/2005 11:20 AM
helpctr.exe	c:\win1828\pchealth\helpctr\binaries\helpctr.exe	2820	8	409600	2826240	11/21/2005 3:08 PM	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	2.01 MB (2,105,344 bytes)	3/30/2005 5:34 PM
wmiprvse.exe	Not Available	672	8	Not Available	Not Available	11/21/2005 3:08 PM	Not Available	Not Available	Not Available

helpsvc.exe c:\win1828\pchealth\helpctr\binaries\helpsvc.exe 2484 8 409600 2826240 11/21/2005 3:08 PM 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.21 MB (2,320,384 bytes) 3/30/2005 5:34 PM

[Loaded Modules]

Name Version Size File Date Manufacturer Path

winlogon 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 664.50 KB (680,448 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winlogon.exe
ntdll 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.56 MB (1,638,400 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntdll.dll
kernel32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.87 MB (1,965,056 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\kernel32.dll
msvcrt 7.0.3790.1828 (srv03_sp1_rtm.050321-2020) 932.00 KB (954,368 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msvcrt.dll
advapi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.41 MB (1,482,752 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\advapi32.dll
rpcrt4 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.34 MB (2,457,600 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\rpcrt4.dll
user32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.41 MB (1,476,096 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\user32.dll
gdi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 882.00 KB (903,168 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\gdi32.dll
userenv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.49 MB (1,563,648 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\userenv.dll
nddeapi 5.2.3790.0 (srv03_rtm.030324-2048) 39.50 KB (40,448 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\nddeapi.dll
crypt32 5.131.3790.1828 (srv03_sp1_rtm.050321-2020) 1.68 MB (1,759,232 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\crypt32.dll
msasn1 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 179.50 KB (183,808 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msasn1.dll
secur32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 186.00 KB (190,464 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\secur32.dll
winsta 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 143.50 KB (146,944 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winsta.dll
netapi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 885.50 KB (906,752 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netapi32.dll
profmap 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 59.50 KB (60,928 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\profmap.dll
regapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 141.50 KB (144,896 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\regapi.dll
ws2_32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 248.00 KB (253,952 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\ws2_32.dll
ws2help 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 51.00 KB (52,224 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\ws2help.dll
msgina 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.40 MB (1,465,344 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msgina.dll
shsvcs 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 354.50 KB (363,008 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\shsvcs.dll
shlwapi 6.00.3790.2440 (srv03_sp1_gdr.050506-1520) 804.00 KB (823,296 bytes) 5/8/2005 11:35 AM Microsoft Corporation c:\win1828\system32\shlwapi.dll
sfc 5.2.3790.0 (srv03_rtm.030324-2048) 7.50 KB (7,680 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sfc.dll
sfc_os 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 273.00 KB (279,552 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sfc_os.dll
wintrust 5.131.3790.1828 (srv03_sp1_rtm.050321-2020) 459.50 KB (470,528 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wintrust.dll
imagehlp 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 140.50 KB (143,872 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\imagehlp.dll
ole32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 3.81 MB (3,992,064 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ole32.dll
comctl32 6.0 (srv03_sp1_rtm.050321-2020) 2.50 MB (2,622,976 bytes) 3/30/2005 7:45 AM Microsoft Corporation c:\win1828\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.3790.1828_x-ww_aa3d7338\comctl32.dll

version 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 51.00 KB (52,224 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\version.dll
winscard 5.2.3790.0 (srv03_rtm.030324-2048) 291.50 KB (298,496 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winscard.dll
wtsapi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 54.00 KB (55,296 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wtsapi32.dll
sxs 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.82 MB (1,904,640 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sxs.dll
shell32 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 12.63 MB (13,243,392 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\shell32.dll
setupapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.99 MB (2,086,400 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\setupapi.dll
wldap32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 441.50 KB (452,096 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wldap32.dll
cscdll 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 207.00 KB (211,968 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cscdll.dll
dimsntfy 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 51.50 KB (52,736 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dimsntfy.dll
wlnotify 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 239.50 KB (245,248 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wlnotify.dll
winmm 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 428.00 KB (438,272 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winmm.dll
winspool 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 400.50 KB (410,112 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winspool.drv
mpr 5.2.3790.0 (srv03_rtm.030324-2048) 163.00 KB (166,912 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mpr.dll
oleaut32 5.2.3790.1828 3.75 MB (3,930,624 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\oleaut32.dll
comctl32 5.82 (srv03_sp1_rtm.050321-2020) 1.72 MB (1,806,336 bytes) 3/30/2005 7:45 AM Microsoft Corporation c:\win1828\winsxs\ia64_microsoft.windows.common-controls_6595b64144ccf1df_5.82.3790.1828_x-ww_4ac92db3\comctl32.dll

uxtheme 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 543.00 KB (556,032 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\uxtheme.dll
scredir 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 69.00 KB (70,656 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\scredir.dll
samlib 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 106.50 KB (109,056 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\samlib.dll
clbcatq 2001.12.4720.1828 (srv03_sp1_rtm.050321-2020) 1.30 MB (1,359,360 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\clbcatq.dll
comres 2001.12.4720.0 (srv03_rtm.030324-2048) 779.50 KB (798,208 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\comres.dll

cscui 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 628.00 KB (643,072 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cscui.dll
ntmarta 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 366.00 KB (374,784 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntmarta.dll
drprov 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 32.50 KB (33,280 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\drprov.dll
rdpsnd 5.2.3790.0 (srv03_rtm.030324-2048) 63.00 KB (64,512 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rdpsnd.dll
psapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 53.00 KB (54,272 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\psapi.dll
ntlanman 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 117.00 KB (119,808 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntlanman.dll
netui0 5.2.3790.0 (srv03_rtm.030324-2048) 181.50 KB (185,856 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netui0.dll
netui1 5.2.3790.0 (srv03_rtm.030324-2048) 482.00 KB (493,568 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netui1.dll
davclnt 5.2.3790.0 (srv03_rtm.030324-2048) 59.00 KB (60,416 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\davclnt.dll
mprui 5.2.3790.0 (srv03_rtm.030324-2048) 97.00 KB (99,328 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mprui.dll
netui2 5.2.3790.0 (srv03_rtm.030324-2048) 761.50 KB (779,776 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netui2.dll
comdlg32 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 745.50 KB (763,392 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\comdlg32.dll
netmsg 5.2.3790.0 (srv03_rtm.030324-2048) 177.50 KB (181,760 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netmsg.dll
msacm32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 87.00 KB (89,088 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msacm32.drv
msacm32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 276.50 KB (283,136 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msacm32.dll
imaadp32 5.2.3790.0 (srv03_rtm.030324-2048) 55.00 KB (56,320 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\imaadp32.acm
msadp32 5.2.3790.0 (srv03_rtm.030324-2048) 49.00 KB (50,176 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msadp32.acm
msg711 5.2.3790.0 (srv03_rtm.030324-2048) 33.00 KB (33,792 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msg711.acm
msgsm32 5.2.3790.0 (srv03_rtm.030324-2048) 66.50 KB (68,096 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msgsm32.acm
tssoft32 1.01 29.00 KB (29,696 bytes) 3/30/2005 3:21 PM DSP GROUP, INC. c:\win1828\system32\tssoft32.acm
tsd32 1.03 38.00 KB (38,912 bytes) 3/30/2005 3:21 PM DSP GROUP, INC. c:\win1828\system32\tsd32.dll
xpsp2res 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.76 MB (2,897,920 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\xpsp2res.dll
services 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 300.00 KB (307,200 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\services.exe
scesrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 806.50 KB (825,856 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\scesrv.dll
authz 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 220.50 KB (225,792 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\authz.dll
umpnpmgr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 323.50 KB (331,264 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\umpnpmgr.dll
ncobjapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 121.00 KB (123,904 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ncobjapi.dll
msvcp60 6.10.2240.8 941.50 KB (964,096 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msvcp60.dll
eventlog 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 180.00 KB (184,320 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\eventlog.dll
lsass 5.2.3790.0 (srv03_rtm.030324-2048) 15.00 KB (15,360 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\lsass.exe
lsasrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.07 MB (2,166,784 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\lsasrv.dll
ntdsapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 204.50 KB (209,408 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntdsapi.dll
dnsapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 428.00 KB (438,272 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dnsapi.dll
samsrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.09 MB (1,142,784 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\samsrv.dll
cryptdll 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 68.50 KB (70,144 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cryptdll.dll
msprivs 5.2.3790.0 (srv03_rtm.030324-2048) 46.00 KB (47,104 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msprivs.dll
kerberos 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 932.00 KB (954,368 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\kerberos.dll
msv1_0 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 380.00 KB (389,120 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msv1_0.dll
iphlpapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 254.50 KB (260,608 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\iphlpapi.dll
netlogon 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 956.00 KB (978,944 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netlogon.dll
w32time 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 565.00 KB (578,560 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\w32time.dll
schannel 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 456.00 KB (466,944 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\schannel.dll
wdigest 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 210.00 KB (215,040 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wdigest.dll
rsaenh 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 415.98 KB (425,960 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\rsaenh.dll
rassfm 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 68.50 KB (70,144 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rassfm.dll
kdcsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 596.00 KB (610,304 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\kdcsvc.dll
ntdsa 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 4.04 MB (4,239,360 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntdsa.dll
esent 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.65 MB (2,776,064 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\esent.dll
ntdsatq 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 79.50 KB (81,408 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntdsatq.dll
mswsock 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 763.00 KB (781,312 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mswsock.dll
scecli 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 459.50 KB (470,528 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\scecli.dll
ws03res 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 792.50 KB (811,520 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\ws03res.dll
hnetcfg 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.04 MB (1,094,144 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\hnetcfg.dll

wshtcpip 5.2.3790.0 (srv03_rtm.030324-2048) 38.00 KB (38,912 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wshtcpip.dll
ipsecsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 465.00 KB (476,160 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ipsecsvc.dll
oakley 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 516.50 KB (528,896 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\oakley.dll
winipsec 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 83.00 KB (84,992 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\winipsec.dll
pstorsvc 5.2.3790.0 (srv03_rtm.030324-2048) 56.00 KB (57,344 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\pstorsvc.dll
psbase 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 167.50 KB (171,520 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\psbase.dll
dssenh 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 363.98 KB (372,712 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dssenh.dll
wlbsctrl 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 207.00 KB (211,968 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wlbsctrl.dll
svchost 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 36.00 KB (36,864 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\svchost.exe
rpcss 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 823.50 KB (843,264 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\rpcss.dll
schedsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 567.00 KB (580,608 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\system32\schedsvc.dll
msidle 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 11.50 KB (11,776 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msidle.dll
wkssvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 315.50 KB (323,072 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wkssvc.dll
wiarpc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 76.00 KB (77,824 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wiarpc.dll
cryptsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 139.00 KB (142,336 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cryptsvc.dll
certcli 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 580.00 KB (593,920 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\certcli.dll
atl 3.00.2282 348.00 KB (356,352 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\atl.dll

vssapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.57 MB (1,642,496 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\vssapi.dll
dmserver 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 52.00 KB (53,248 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dmserver.dll
es 2001.12.4720.1828 (srv03_sp1_rtm.050321-2020) 685.50 KB (701,952 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\es.dll
srvsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 197.50 KB (202,240 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\srvsvc.dll
netman 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 706.50 KB (723,456 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netman.dll
netshell 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.95 MB (3,094,528 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netshell.dll
rtutils 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 91.50 KB (93,696 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\rtutils.dll
credui 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 311.00 KB (318,464 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\credui.dll
clusapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 158.00 KB (161,792 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\clusapi.dll
mprapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 265.00 KB (271,360 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mprapi.dll
activeds 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 581.50 KB (595,456 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\activeds.dll
adsldpc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 326.00 KB (333,824 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\adsldpc.dll
rasapi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 632.50 KB (647,680 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rasapi32.dll
rasman 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 170.00 KB (174,080 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rasman.dll
tapi32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 518.00 KB (530,432 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\tapi32.dll
wzcsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 818.00 KB (837,632 bytes) 3/21/2005 3:29 PM Microsoft Corporation c:\win1828\system32\wzcsvc.dll
wmi 5.2.3790.0 (srv03_rtm.030324-2048) 5.00 KB (5,120 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wmi.dll
dhcpcsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 299.50 KB (306,688 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dhcpcsvc.dll

wininet 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 1.62 MB (1,697,280 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wininet.dll
wzcsapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 86.50 KB (88,576 bytes) 3/21/2005 3:29 PM Microsoft Corporation c:\win1828\system32\wzcsapi.dll
sacsvr 5.2.3790.0 (srv03_rtm.030324-2048) 27.50 KB (28,160 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sacsvr.dll
sens 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 98.00 KB (100,352 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sens.dll
trkwks 5.2.3790.0 (srv03_rtm.030324-2048) 246.00 KB (251,904 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\trkwks.dll
wmisvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 445.50 KB (456,192 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wmisvc.dll
ipnathlp 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 945.50 KB (968,192 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ipnathlp.dll
browser 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 200.50 KB (205,312 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\browser.dll
rasadhlp 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 18.00 KB (18,432 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rasadhlp.dll
wbemcomn 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 665.50 KB (681,472 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wbem\wbemcomn.dll
wbemcore 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.94 MB (2,038,784 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wbemcore.dll
esscli 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.01 MB (1,057,280 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\esscli.dll
fastprox 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.63 MB (1,710,592 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\fastprox.dll
wbemsvc 5.2.3790.0 (srv03_rtm.030324-2048) 62.50 KB (64,000 bytes) 3/30/2005 5:33 PM Microsoft Corporation c:\win1828\system32\wbem\wbemsvc.dll
wmiutils 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 302.00 KB (309,248 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 649.00 KB (664,576 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.39 MB (1,454,592 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.02 MB (1,070,080 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wbemess.dll

ncprov 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 146.00 KB (149,504 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\ncprov.dll
netrap 5.2.3790.0 (srv03_rtm.030324-2048) 30.00 KB (30,720 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\netrap.dll
rasdlg 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.44 MB (1,509,888 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rasdlg.dll
pchsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 113.00 KB (115,712 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\pchsvc.dll
wbemcons 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 154.50 KB (158,208 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wbemcons.dll
xactsrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 297.00 KB (304,128 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\xactsrv.dll
ersvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 66.50 KB (68,096 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ersvc.dll
termsrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 680.00 KB (696,320 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\termsrv.dll
icaapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 33.50 KB (34,304 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\icaapi.dll
mstlsapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 340.50 KB (348,672 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mstlsapi.dll
rdpwsx 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 325.13 KB (332,936 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\rdpwsx.dll
sacsess 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 84.00 KB (86,016 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sacsess.exe
rdpclip 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 211.00 KB (216,064 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\rdpclip.exe
wsock32 5.2.3790.0 (srv03_rtm.030324-2048) 23.00 KB (23,552 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wsock32.dll
urlmon 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 1.53 MB (1,600,000 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\urlmon.dll
explorer 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 1.64 MB (1,720,320 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\explorer.exe
browseui 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 2.42 MB (2,542,592 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\browseui.dll

shdocvw 6.00.3790.2440 (srv03_sp1_gdr.050506-1520) 3.49 MB (3,658,752 bytes) 5/8/2005 11:35 AM Microsoft Corporation c:\win1828\system32\shdocvw.dll
cryptui 5.131.3790.1828 (srv03_sp1_rtm.050321-2020) 1.11 MB (1,159,168 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cryptui.dll
apphelp 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 301.50 KB (308,736 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\apphelp.dll
themeui 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 891.00 KB (912,384 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\themeui.dll
msimg32 5.2.3790.0 (srv03_rtm.030324-2048) 7.00 KB (7,168 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msimg32.dll
linkinfo 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 46.50 KB (47,616 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\linkinfo.dll
ntshrui 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 244.00 KB (249,856 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntshrui.dll
msi 3.1.4000.1828 5.71 MB (5,984,768 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msi.dll
webcheck 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 699.00 KB (715,776 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\webcheck.dll
stobject 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 175.00 KB (179,200 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\stobject.dll
batmeter 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 61.00 KB (62,464 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\batmeter.dll
powrprof 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 44.00 KB (45,056 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\powrprof.dll
browselc 6.00.3790.0 (srv03_rtm.030324-2048) 61.50 KB (62,976 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\browselc.dll
duser 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 756.00 KB (774,144 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\duser.dll
mlang 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 813.50 KB (833,024 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mlang.dll
actxprxy 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 237.00 KB (242,688 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\actxprxy.dll
mmc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.51 MB (2,631,168 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mmc.exe
mfc42u 6.50.4245.0 3.35 MB (3,510,272 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mfc42u.dll

oleacc 4.2.5406.0 (srv03_rtm.030324-2048) 485.00 KB (496,640 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\oleacc.dll
mmcbase 5.2.3790.0 (srv03_rtm.030324-2048) 139.00 KB (142,336 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mmcbase.dll
mmcndmgr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 3.18 MB (3,331,584 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mmcndmgr.dll
msxml3 8.70.1104.0 3.33 MB (3,488,768 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msxml3.dll

cmprops 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 492.00 KB (503,808 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\cmprops.dll
mmfutil 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 28.00 KB (28,672 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\mmfutil.dll
ntmsmgr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.21 MB (1,268,224 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntmsmgr.dll
ntmsapi 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 128.00 KB (131,072 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\ntmsapi.dll
els 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 520.50 KB (532,992 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\els.dll
dfrgsnap 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 107.00 KB (109,568 bytes) 3/30/2005 3:20 PM Microsoft Corp. and Executive Software International, Inc. c:\win1828\system32\dfrgsnap.dll

dfrgres 5.2.3790.0 (srv03_rtm.030324-2048) 50.00 KB (51,200 bytes) 3/30/2005 3:20 PM Microsoft Corp. and Executive Software International, Inc. c:\win1828\system32\dfrgres.dll

mycomput 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 267.50 KB (273,920 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mycomput.dll
filemgmt 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 928.00 KB (950,272 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\filemgmt.dll
cfgmgr32 5.2.3790.0 (srv03_rtm.030324-2048) 16.00 KB (16,384 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cfgmgr32.dll
wbemcntl 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 549.50 KB (562,688 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wbemcntl.dll
localsec 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 644.50 KB (659,968 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\localsec.dll
smlogcfg 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.01 MB (1,056,256 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\smlogcfg.dll

pdh 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 807.00 KB (826,368 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\pdh.dll
odbc32 3.525.1014.0620.00 KB (634,880 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\odbc32.dll

odbcbcp 2000.085.1014.00 48.00 KB (49,152 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\odbcbcp.dll
odbcint 3.525.1014.088.00 KB (90,112 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\odbcint.dll

snmpsnap 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 565.00 KB (578,560 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\snmpsnap.dll
dmdskmgr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 542.00 KB (555,008 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dmdskmgr.dll
dmutil 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 73.00 KB (74,752 bytes) 3/21/2005 3:12 PM Microsoft Corporation c:\win1828\system32\dmutil.dll
dmdskres 5.2.3790.0 (srv03_rtm.030324-2048) 115.00 KB (117,760 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dmdskres.dll
devmgr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 628.00 KB (643,072 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\devmgr.dll
rasuser 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 513.50 KB (525,824 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\rasuser.dll
dsprop 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 361.50 KB (370,176 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dsprop.dll
dsuiext 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 241.00 KB (246,784 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dsuiext.dll
mprsnap 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.84 MB (2,979,328 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mprsnap.dll
rtrfiltr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 240.00 KB (245,760 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\rtrfiltr.dll
servdeps 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 151.50 KB (155,136 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\servdeps.dll
riched32 5.2.3790.0 (srv03_rtm.030324-2048) 5.00 KB (5,120 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\riched32.dll
riched20 5.31.23.12241.38 MB (1,451,008 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\riched20.dll

adsnt 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 753.50 KB (771,584 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\adsnt.dll
dmocx 5.2.3790.0 (srv03_rtm.030324-2048) 58.00 KB (59,392 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\dmocx.dll
cryptnet 5.131.3790.1828 (srv03_sp1_rtm.050321-2020) 168.50 KB (172,544 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cryptnet.dll
sensapi 5.2.3790.0 (srv03_rtm.030324-2048) 11.50 KB (11,776 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\sensapi.dll
cabinet 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 250.50 KB (256,512 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\cabinet.dll
imm32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 322.50 KB (330,240 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\imm32.dll
lputilnt 1,8,a,11 472.00 KB (483,328 bytes) 4/14/2005 11:20 AM c:\program files (x86)\emulex\lputilnt.exe

wow64 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 487.00 KB (498,688 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wow64.dll
wow64win 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 567.50 KB (581,120 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wow64win.dll
wow64cpu 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 44.00 KB (45,056 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\wow64cpu.dll
wowia32x 5,3,5021,0 54.00 KB (55,296 bytes) 3/21/2005 3:29 PM Intel Corporation c:\win1828\system32\wowia32x.dll

ia32exec 5,3,5337,0 7.13 MB (7,471,104 bytes) 3/21/2005 3:15 PM Intel Corporation c:\win1828\system32\ia32exec.bin

helpctr 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.01 MB (2,105,344 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\helpctr.exe
hcappres 5.2.3790.0 (srv03_rtm.030324-2048) 6.00 KB (6,144 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\hcappres.dll
itss 5.2.3790.2427 (srv03_sp1_gdr.050421-1629) 411.00 KB (420,864 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\itss.dll
pchshell 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 290.50 KB (297,472 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\pchshell.dll
mshtml 6.00.3790.2440 (srv03_sp1_gdr.050506-1520) 8.85 MB (9,283,072 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mshtml.dll
msls31 3.10.349.0 502.50 KB (514,560 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\msls31.dll

msimtf 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 586.50 KB (600,576 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\msimtf.dll
msctf 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 983.50 KB (1,007,104 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\msctf.dll
shdoclc 6.00.3790.0 (srv03_rtm.030324-2048) 588.00 KB (602,112 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\shdoclc.dll
jscript 5.6.0.8827 1.24 MB (1,304,576 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\jscript.dll

mshtmled 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 1.46 MB (1,531,392 bytes) 3/30/2005 3:20 PM Microsoft Corporation c:\win1828\system32\mshtmled.dll
vbscript 5.6.0.8827 1.07 MB (1,118,208 bytes) 3/30/2005 3:21 PM Microsoft Corporation c:\win1828\system32\vbscript.dll

msinfo 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 1.26 MB (1,321,472 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\msinfo.dll
wbemprox 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 65.00 KB (66,560 bytes) 3/30/2005 5:32 PM Microsoft Corporation c:\win1828\system32\wbem\wbemprox.dll
helpsvc 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.21 MB (2,320,384 bytes) 3/30/2005 5:34 PM Microsoft Corporation c:\win1828\pchealth\helpctr\binaries\helpsvc.exe

[Services]

Display Name Name State Start Mode Service Type Path Error Control Start Name Tag ID
Application Experience Lookup Service AeLookupSvc Stopped Disabled Share Process c:\win1828\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Alerter Alerter Running Auto Share Process c:\win1828\system32\svchost.exe -k localservice Normal NT AUTHORITY\LocalService 0
Application Layer Gateway Service ALG Running Manual Own Process c:\win1828\system32\alg.exe Normal NT AUTHORITY\LocalService 0
Application Management AppMgmt Stopped Manual Share Process c:\win1828\system32\svchost.exe -k netsvcs Normal LocalSystem 0
ASP.NET State Service aspnet_state Stopped Manual Own Process c:\win1828\microsoft.net\framework64\v2.0.50727\aspnet_state.exe Normal NT AUTHORITY\NetworkService 0

Windows Audio AudioSrv Stopped Disabled Share Process c:\win1828\system32\svchost.exe -k netsvcs Normal LocalSystem 0

```
Background Intelligent Transfer Service BITS Stopped Manual
        Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Computer Browser Browser Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Indexing Service CiSvc Stopped Disabled Share
Process c:\win1828\system32\cisvc.exe Normal
        LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
        c:\win1828\system32\clipsrv.exe Normal
        LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_X86
        clr_optimization_v2.0.50727_32 Stopped Manual
        Own Process
        c:\win1828\microsoft.net\framework\v2.0.50727\mscorsvw.exe
        Ignore LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_I64
        clr_optimization_v2.0.50727_64 Stopped Manual
        Own Process
        c:\win1828\microsoft.net\framework64\v2.0.50727\mscorsvw.e
xe Ignore LocalSystem 0
COM+ System Application COMSysApp Stopped Manual Own
Process c:\win1828\system32\dllhost.exe /processid:{02d4b3f1-fd88-
11d1-960d-00805fc79235} Normal LocalSystem 0
Cryptographic Services CryptSvc Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
DCOM Server Process Launcher DcomLaunch Running Auto
        Share Process c:\win1828\system32\svchost.exe -k
dcomlaunch Normal LocalSystem 0
Distributed File System Dfs Stopped Manual Own
Process c:\win1828\system32\dfssvc.exe Normal
        LocalSystem 0
DHCP Client Dhcp Running Auto Share Process
        c:\win1828\system32\svchost.exe -k networkservice Normal
        NT AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service dmadmin Stopped
        Manual Share Process
        c:\win1828\system32\dmadmin.exe /com Normal
        LocalSystem 0
Logical Disk Manager dmserver Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
DNS Client Dnscache Running Auto Share Process
        c:\win1828\system32\svchost.exe -k networkservice Normal
        NT AUTHORITY\NetworkService 0
Emulex HBAnyware SvcMgr Emulex HBAnyware SvcMgr
        Stopped Manual Own Process "c:\program files
(x86)\hbanyware\hbahsmgr.exe" Normal LocalSystem 0
```

```
Error Reporting Service ERSvc Running Auto Share
Process c:\win1828\system32\svchost.exe -k winerr Ignore
Event Log Eventlog Running Auto Share Process
        c:\win1828\system32\services.exe Normal
        LocalSystem 0
COM+ Event System EventSystem Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Help and Support helpsvc Running Manual Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Human Interface Device Access HidServ Stopped Disabled
        Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
HTTP SSL HTTPFilter Stopped Manual Share Process
        c:\win1828\system32\lsass.exe Normal
        LocalSystem 0
IAS Jet Database Access IASJet Stopped Manual Share
Process c:\win1828\syswow64\svchost.exe -k iasjet Normal
        LocalSystem 0
IMAPI CD-Burning COM Service ImapiService Stopped Disabled
        Own Process c:\win1828\system32\imapi.exe Normal
        LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own
Process c:\win1828\system32\ismserv.exe Normal
        LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled
        Share Process c:\win1828\system32\lsass.exe
        Normal LocalSystem 0
Server lanmanserver Running Auto Share Process
        c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Workstation lanmanworkstation Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
License Logging LicenseService Stopped Disabled
        Own Process c:\win1828\system32\llssrv.exe Normal
        NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Running Auto Share
Process c:\win1828\system32\svchost.exe -k localservice Normal
        NT AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
        c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Disabled
        Own Process c:\win1828\system32\mnmsrvc.exe Normal
        LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto
        Own Process c:\win1828\system32\msdtc.exe Normal
        NT AUTHORITY\NetworkService 0
```

```
SQL Server Integration Services MsDtsServer Stopped Disabled
        Own Process "c:\program files\microsoft sql
server\90\dts\binn\msdtssrvr.exe" Normal NT
AUTHORITY\NetworkService 0
SQL Server Full Text Search (MSSQLSERVER) msftesql Stopped
        Disabled Own Process "c:\program files\microsoft sql
server\mssql.1\mssql\binn\msftesql.exe" -s:mssql.1 -f:mssqlserver Normal
        LocalSystem 0
Windows Installer MSIServer Stopped Manual Share
Process c:\win1828\system32\msiexec.exe /v Normal
        LocalSystem 0
Microsoft Search MSSEARCH Stopped Auto Share
Process "c:\program files\common
files\system\mssearch\bin\mssearch.exe" Normal
        LocalSystem 0
SQL Server (MSSQLSERVER) MSSQLSERVER Stopped
        Manual Own Process "c:\program files\microsoft sql
server\mssql.1\mssql\binn\sqlservr.exe" -smssqlserver Normal
        LocalSystem 0
SQL Server Active Directory Helper MSSQLServerADHelper Stopped
        Disabled Own Process "c:\program files\microsoft sql
server\90\shared\sqladhlp90.exe" Normal NT
AUTHORITY\NetworkService 0
SQL Server Analysis Services (MSSQLSERVER)
        MSSQLServerOLAPService Stopped Manual Own
Process "c:\program files\microsoft sql
server\mssql.2\olap\bin\msmdsrv.exe" -s "c:\program files\microsoft sql
server\mssql.2\olap\config" Normal LocalSystem 0

Network DDE NetDDE Stopped Disabled Share Process
        c:\win1828\system32\netdde.exe Normal
        LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Disabled Share
Process c:\win1828\system32\netdde.exe Normal
        LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process
        c:\win1828\system32\lsass.exe Normal
        LocalSystem 0
Network Connections Netman Running Manual Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
Network Location Awareness (NLA) Nla Running Manual
        Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own
Process c:\win1828\system32\ntfrs.exe Ignore
        LocalSystem 0
NT LM Security Support Provider NtLmSsp Running Manual
        Share Process c:\win1828\system32\lsass.exe
        Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Manual Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
        LocalSystem 0
```

```
Office Source Engine ose Stopped Manual Own
Process "c:\program files (x86)\common files\microsoft shared\source
engine\ose.exe" Normal LocalSystem 0
Plug and PlayPlugPlay Running Auto Share Process
c:\win1828\system32\services.exe Normal
LocalSystem 0
IPSEC Services PolicyAgent Running Auto Share
Process c:\win1828\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Running Auto
Share Process c:\win1828\system32\lsass.exe
Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped
Manual Share Process
c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr Stopped Manual
Own Process c:\win1828\system32\sessmgr.exe Normal
LocalSystem 0
Routing and Remote Access RemoteAccess Stopped
Disabled Share Process
c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Registry RemoteRegistry Stopped Disabled
Share Process c:\win1828\system32\svchost.exe -k
regsvc Normal NT AUTHORITY\LocalService 0

Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual
Own Process c:\win1828\system32\locator.exe Normal
NT AUTHORITY\NetworkService 0
Remote Procedure Call (RPC) RpcSs Running Auto
Share Process c:\win1828\system32\svchost.exe -k
rpcss Normal NT AUTHORITY\NetworkService 0

Resultant Set of Policy Provider RSoPProv Stopped Manual
Share Process c:\win1828\system32\rsopprov.exe
Normal LocalSystem 0
Special Administration Console Helper sacsvr Running Manual
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share
Process c:\win1828\system32\lsass.exe Normal
LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
c:\win1828\system32\scardsvr.exe Ignore NT
AUTHORITY\LocalService 0
Task Scheduler Schedule Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0

Secondary Logon seclogon Stopped Disabled Share
Process c:\win1828\system32\svchost.exe -k netsvcs Ignore
System Event Notification SENS Running Auto Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Windows Firewall/Internet Connection Sharing (ICS) SharedAccess
Running Auto Share Process
c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Shell Hardware Detection ShellHWDetection Running Auto
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Print Spooler Spooler Stopped Disabled Own Process
c:\win1828\system32\spoolsv.exe Normal
LocalSystem 0
SQL Server Browser SQLBrowser Stopped Disabled Own
Process "c:\program files\microsoft sql
server\90\shared\sqlbrowser.exe" Normal LocalSystem 0

SQL Server Agent (MSSQLSERVER) SQLSERVERAGENT Stopped
Manual Own Process "c:\program files\microsoft sql
server\mssql.1\mssql\binn\sqlagent90.exe" -i mssqlserver Normal
LocalSystem 0
SQL Server VSS Writer SQLWriter Stopped Manual Own
Process "c:\program files\microsoft sql server\90\shared\sqlwriter.exe"
Normal LocalSystem 0
Windows Image Acquisition (WIA) stisvc Stopped Disabled
Share Process c:\win1828\system32\svchost.exe -k
imgsvc Normal NT AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider swprv Stopped
Manual Own Process c:\win1828\system32\svchost.exe -k
swprv Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Auto
Own Process c:\win1828\system32\smlogsvc.exe Normal
NT Authority\NetworkService 0
Telephony TapiSrv Stopped Manual Share Process
c:\win1828\system32\svchost.exe -k tapisrv Normal
LocalSystem 0
Terminal Services TermService Running Manual Share
Process c:\win1828\system32\svchost.exe -k termsvcs Normal
LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Distributed Link Tracking Server TrkSvr Stopped Disabled
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0

Terminal Services Session Directory Tssdis Stopped Disabled
Own Process c:\win1828\system32\tssdis.exe Normal
Uninterruptible Power Supply UPS Stopped Manual
Own Process c:\win1828\system32\ups.exe Normal
NT AUTHORITY\LocalService 0
Virtual Disk Service vds Stopped Manual Own
Process c:\win1828\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\win1828\system32\vssvc.exe Normal
LocalSystem 0
Windows Time W32Time Running Auto Share
Process c:\win1828\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
WebClient WebClient Stopped Disabled Share Process
c:\win1828\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service WinHttpAutoProxySvc
Stopped Manual Share Process
c:\win1828\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt Running Auto
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi
Stopped Manual Share Process
c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped Manual Own
Process c:\win1828\system32\wbem\wmiapsrv.exe Normal
LocalSystem 0
Automatic Updates wuauserv Stopped Disabled Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Wireless Configuration WZCSVC Stopped Disabled Share
Process c:\win1828\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Network Provisioning Service xmlprov Stopped Manual
Share Process c:\win1828\system32\svchost.exe -k
netsvcs Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default
User
Accessories\Entertainment Default User:Accessories\Entertainment Default
User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users
```

Accessories\Communications All
Users:Accessories\Communications All Users
Accessories\Entertainment All Users:Accessories\Entertainment All Users

Accessories\System Tools All Users:Accessories\System Tools All Users

ActiveState ActivePerl 5.6 All Users:ActiveState ActivePerl 5.6 All Users

Administrative Tools All Users:Administrative Tools All Users

Debugging Tools for Windows 64-bit All Users:Debugging Tools for
Windows 64-bit All Users
Emulex All Users:Emulex All Users
HP SAN Utilities All Users:HP SAN Utilities All Users
Java Web Start All Users:Java Web Start All Users
Microsoft SQL Server 2005 CTP All Users:Microsoft SQL Server 2005
CTP All Users
Microsoft SQL Server 2005 CTP\Configuration Tools All Users:Microsoft SQL
Server 2005 CTP\Configuration Tools All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories NT
AUTHORITY\SYSTEM
Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility
NT AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories SQLORCA\Administrator:Accessories
SQLORCA\Administrator
Accessories\Accessibility SQLORCA\Administrator:Accessories\Accessibility
SQLORCA\Administrator
Accessories\Entertainment
SQLORCA\Administrator:Accessories\Entertainment
SQLORCA\Administrator
Administrative Tools SQLORCA\Administrator:Administrative Tools
SQLORCA\Administrator
HP SAN Utilities SQLORCA\Administrator:HP SAN Utilities
SQLORCA\Administrator
Microsoft SQL Server 2005 CTP SQLORCA\Administrator:Microsoft
SQL Server 2005 CTP SQLORCA\Administrator
Microsoft SQL Server 2005 CTP\Configuration Tools
SQLORCA\Administrator:Microsoft SQL Server 2005
CTP\Configuration Tools SQLORCA\Administrator
QLogic Corporation SQLORCA\Administrator:QLogic Corporation
SQLORCA\Administrator
QLogic Corporation\SANblade Control VIX
SQLORCA\Administrator:QLogic Corporation\SANblade Control
VIX SQLORCA\Administrator
Startup SQLORCA\Administrator:Startup
SQLORCA\Administrator

[Startup Programs]

Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM Startup
desktop desktop.ini SQLORCA\Administrator Startup
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common Startup

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
WordPad Document "%programfiles%\windows
nt\accessories\wordpad.exe"
Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details
11/21/2005 9:38 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/21/2005 9:36 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/21/2005 9:33 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/18/2005 1:17 PM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/18/2005 1:11 PM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/18/2005 10:32 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/18/2005 10:32 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

11/18/2005 10:18 AM Application Error Faulting application
sleep.exe, version 0.0.0.0, faulting module sleep.exe, version 0.0.0.0, fault
address 0x0000566f.

[Internet Settings]

[Internet Explorer]
[Following are sub-categories of this main category]
[Summary]

Item Value
Version 6.0.3790.1828
Build 63790.1828
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active PrinterNot Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company

actxprxy.dll 6.0.3790.1828 237 KB 3/21/2005 11:32:37
PM C:\WIN1828\system32 Microsoft Corporation
advpack.dll 6.0.3790.1828 260 KB 3/21/2005 11:32:38
PM C:\WIN1828\system32 Microsoft Corporation
asctrls.ocx 6.0.3790.0 219 KB 3/25/2003 1:18:37 PM
C:\WIN1828\system32 Microsoft Corporation
browselc.dll 6.0.3790.0 62 KB 3/25/2003 1:16:19 PM
C:\WIN1828\system32 Microsoft Corporation
browseui.dll 6.0.3790.1828 2,483 KB 3/21/2005 11:32:43
PM C:\WIN1828\system32 Microsoft Corporation
cdfview.dll 6.0.3790.1828 307 KB 3/21/2005 11:32:44
PM C:\WIN1828\system32 Microsoft Corporation
comctl32.dll 5.82.3790.1828 1,764 KB 3/21/2005 11:32:47
PM C:\WIN1828\system32 Microsoft Corporation
dxtrans.dll 6.3.3790.1828 618 KB 3/21/2005 11:32:57
PM C:\WIN1828\system32 Microsoft Corporation
dxtmsft.dll 6.3.3790.1828 986 KB 3/21/2005 11:32:57
PM C:\WIN1828\system32 Microsoft Corporation
iecont.dll <File Missing> Not Available Not Available Not
Available Not Available
iecontlc.dll <File Missing> Not Available Not Available Not
Available Not Available
iedkcs32.dll 16.0.3790.1828 702 KB 3/21/2005 11:33:04
PM C:\WIN1828\system32 Microsoft Corporation
iepeers.dll 6.0.3790.2440 701 KB 5/8/2005 10:35:26 AM
C:\WIN1828\system32 Microsoft Corporation
iesetup.dll 6.0.3790.1828 98 KB 3/21/2005 11:33:05
PM C:\WIN1828\system32 Microsoft Corporation
ieuinit.inf Not Available 24 KB 3/21/2005 9:02:27 PM
C:\WIN1828\system32 Not Available
iexplore.exe 6.0.3790.1828 108 KB 3/21/2005 9:59:45 PM
C:\Program Files\Internet Explorer Microsoft Corporation

imgutil.dll 6.0.3790.1828 114 KB 3/21/2005 11:33:06 PM C:\WIN1828\system32 Microsoft Corporation
inetcpl.cpl 6.0.3790.1828 666 KB 3/21/2005 11:33:06 PM C:\WIN1828\system32 Microsoft Corporation
inetcplc.dll 6.0.3790.0 108 KB 3/25/2003 1:17:29 PM C:\WIN1828\system32 Microsoft Corporation
inseng.dll 6.0.3790.1828 276 KB 3/21/2005 11:33:06 PM C:\WIN1828\system32 Microsoft Corporation
mlang.dll 6.0.3790.1828 814 KB 3/21/2005 11:33:11 PM C:\WIN1828\system32 Microsoft Corporation
msencode.dll <File Missing> Not Available Not Available Not Available Not Available
mshta.exe 6.0.3790.1828 68 KB 3/21/2005 9:56:54 PM C:\WIN1828\system32 Microsoft Corporation
mshtml.dll 6.0.3790.2440 9,066 KB 5/8/2005 10:35:26 AM C:\WIN1828\system32 Microsoft Corporation
mshtml.tlb 6.0.3790.1828 1,319 KB 3/21/2005 8:57:20 PM C:\WIN1828\system32 Microsoft Corporation
mshtmled.dll 6.0.3790.1828 1,496 KB 3/21/2005 11:33:17 PM C:\WIN1828\system32 Microsoft Corporation
mshtmler.dll 6.0.3790.1828 56 KB 3/21/2005 8:57:24 PM C:\WIN1828\system32 Microsoft Corporation
msident.dll 6.0.3790.1828 144 KB 3/21/2005 11:33:18 PM C:\WIN1828\system32 Microsoft Corporation
msidntld.dll 6.0.3790.0 14 KB 3/25/2003 1:17:49 PM C:\WIN1828\system32 Microsoft Corporation
msieftp.dll 6.0.3790.1828 579 KB 3/21/2005 11:33:18 PM C:\WIN1828\system32 Microsoft Corporation
msrating.dll 6.0.3790.1828 415 KB 3/21/2005 11:33:20 PM C:\WIN1828\system32 Microsoft Corporation
mstime.dll 6.0.3790.1828 1,800 KB 3/21/2005 11:33:20 PM C:\WIN1828\system32 Microsoft Corporation
occache.dll 6.0.3790.1828 218 KB 3/21/2005 11:33:24 PM C:\WIN1828\system32 Microsoft Corporation
proctexe.ocx <File Missing> Not Available Not Available Not Available Not Available
sendmail.dll 6.0.3790.1828 109 KB 3/21/2005 11:33:33 PM C:\WIN1828\system32 Microsoft Corporation
shdoclc.dll 6.0.3790.0 588 KB 3/25/2003 8:45:14 AM C:\WIN1828\system32 Microsoft Corporation
shdocvw.dll 6.0.3790.2440 3,573 KB 5/8/2005 10:35:26 AM C:\WIN1828\system32 Microsoft Corporation
shfolder.dll 6.0.3790.1828 43 KB 3/21/2005 11:33:34 PM C:\WIN1828\system32 Microsoft Corporation
shlwapi.dll 6.0.3790.2440 804 KB 5/8/2005 10:35:26 AM C:\WIN1828\system32 Microsoft Corporation
tdc.ocx 1.3.0.3130 177 KB 3/25/2003 1:18:38 PM C:\WIN1828\system32 Microsoft Corporation
url.dll 6.0.3790.1828 49 KB 3/21/2005 11:33:37 PM C:\WIN1828\system32 Microsoft Corporation
urlmon.dll 6.0.3790.1828 1,563 KB 3/21/2005 11:33:38 PM C:\WIN1828\system32 Microsoft Corporation
webcheck.dll 6.0.3790.1828 699 KB 3/21/2005 11:33:38 PM C:\WIN1828\system32 Microsoft Corporation
wininet.dll 6.0.3790.1828 1,658 KB 3/21/2005 11:33:38 PM C:\WIN1828\system32 Microsoft Corporation

[Connectivity]

Item	Value
Connection Preference	Never dial

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Enabled
ProxyServer	http://itgproxy:80
ProxyOverride	<local>

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	C:\Documents and Settings\ORGANIZA-EZAQT0\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	Not Available
Available Disk Space	Not Available
Maximum Cache Size	Not Available
Available Cache Size	Not Available

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranet Custom
Trusted sites Custom
Internet Custom
Restricted sites Custom

Microsoft SQL Server 2005 Startup Parameters

sqlservr.exe" -c -x -T661 -T677 -T827 -T834 -T3502 -T8011 -T8012 -T8018 -T8020 -T8193 -T8710 -T8744
where
rem ** -c - Run as console app
rem ** -x - Disable stats
rem ** -T661 - Disable the ghost record removal process
rem ** -T677 - Disable bunch of checks in AM. Available in chk/fre, not golden
rem ** -T827 - Display latch promotions
rem ** -T834 - Large pages
rem ** -T3502 - Send checkpoint state changes to errorlog
rem ** -T3605 - Send trace output to the errorlog
rem ** -T8011 - Disable ring buffer for resource monitor
rem ** -T8012 - Disable ring buffer for schedulers
rem ** -T8018 - Disable exception ring buffer
rem ** -T8020 - Disable working set trimming
rem ** -T8193 - Enable large pages support
rem ** -T8710 - disable HP spools (2000 vs 1w bug)
rem ** -T8744 - Disable pre-fetch
rem ** -T652 - Disable readahead

Note: DBCC TRACEON(652,-1) was executed after steady state was achieved. Readahead speeds up rampup, but helps performance after rampup.

Microsoft SQL Server VIA and Soft Numa Config

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server]
"SqlMdacRegRefCount"=dword:00000003
"SsrpActiveServer"=""
"InstalledInstances"=hex(7):4d,00,53,00,53,00,51,00,4c,00,53,00,45,00,52,0
0,56,\
  00,45,00,52,00,00,00,00,00
"IsListenerActive"="No"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90]
"SharedCode"="C:\\Program Files\\Microsoft SQL Server\\90\\Shared\\"
"VerSpecificRootDir"="C:\\Program Files\\Microsoft SQL Server\\90\\"
"EnableErrorReporting"=dword:00000000
"ErrorReportingDir"="C:\\Program Files\\Microsoft SQL
Server\\90\\Shared\\ErrorDumps\\"
"CustomerFeedback"=dword:00000000
"TimeOfReporting"=dword:000000f0
"CollectorParameters"=""


Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node0]
"CpuMask"=hex:01

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node1]
"CpuMask"=hex:02

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node10]
"CpuMask"=hex:00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node11]
"CpuMask"=hex:00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node12]
"CpuMask"=hex:00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node13]
"CpuMask"=hex:00,00,00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node14]
"CpuMask"=hex:00,00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node15]
"CpuMask"=hex:00,00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node16]
"CpuMask"=hex:00,00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node17]
"CpuMask"=hex:00,00,00,00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node18]
"CpuMask"=hex:00,00,00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node19]
"CpuMask"=hex:00,00,00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node2]
"CpuMask"=hex:0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node20]
"CpuMask"=hex:00,00,00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node21]
"CpuMask"=hex:00,00,00,00,00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node22]
"CpuMask"=hex:00,00,00,00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node23]
"CpuMask"=hex:00,00,00,00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node24]
"CpuMask"=hex:00,00,00,00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node25]
"CpuMask"=hex:00,00,00,00,00,00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node26]
"CpuMask"=hex:00,00,00,00,00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node27]
"CpuMask"=hex:00,00,00,00,00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node28]
"CpuMask"=hex:00,00,00,00,00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node29]
"CpuMask"=hex:00,00,00,00,00,00,00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node3]
"CpuMask"=hex:30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node30]
"CpuMask"=hex:00,00,00,00,00,00,00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node31]
"CpuMask"=hex:00,00,00,00,00,00,00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node32]
"CpuMask"=hex:00,00,00,00,00,00,00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node4]
"CpuMask"=hex:c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node5]
"CpuMask"=hex:00,03

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node6]
"CpuMask"=hex:00,0c

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node7]
"CpuMask"=hex:00,30

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node8]
"CpuMask"=hex:00,c0

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node9]
```

```
"CpuMask"=hex:00,00,03
```

U12ser Rights Assignment

The Group Policy Editor of Windows.net was used to modify an entry under User Rights Assignment. Specifically, the right to "Lock pages in memory" was given to the Administrators group so that SQL Server 2000 could use large amounts of physical memory.

LP1050 Driver Settings

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000027
"Type"=dword:00000001
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00
,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,65,00,6c,00,78,00,73,00,74,00,6f
,\
  00,72,00,2e,00,73,00,79,00,73,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Parame
ters]
"BusType"=dword:00000006

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Parame
ters\Device]
"DriverParameter"="CoalesceRspCnt=1;NodeTimeOut=10;LinkTimeOut=40;"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Parame
ters\PnpInterface]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\elxstor\Enum]
"0"="PCI\\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\\5&32c61491
&0&08"
"Count"=dword:00000004
"NextInstance"=dword:00000004
"1"="PCI\\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\\5&3815fdfd
&0&08"
"2"="PCI\\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\\5&36a998ee
&0&08"
"3"="PCI\\VEN_10DF&DEV_F0A5&SUBSYS_F0A510DF&REV_01\\5&24e12db3
&0&08"
```

SmartArray Driver Settings

NOTE: MinimumCosPollingInterval was set to #150 for performance run.

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00
,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,68,00,70,00,71,00,63,00,69,00,7
3,\
  00,73,00,62,00,2e,00,73,00,79,00,73,00,00,00
"DisplayName"="Smart Array Controllers Non-Miniport Bus Driver"
"Group"="port"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Para
meters]
"CompletionMode"=dword:00000002
"EnableInterruptWatchdogTimer"=dword:00000001
"MinimumCosPollingInterval"=dword:00000005
"PerCpuDpcImportance"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Secur
ity]
"Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,0
0,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,
\

00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00
,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,
\
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,04,00,00,00,0
0,\
00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,06,00,00,00,00,00,14,00,00,0
1,\

00,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,00,18,00,fd,01,02,00,01,02,00
,\

00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,12,00,00,0
0,\
  01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Enum
]
"0"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&1457b150
&0&2008"
"Count"=dword:0000003c
"NextInstance"=dword:0000003c
"1"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&1457b150
&0&2808"
"2"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&2b186594
&0&2008"
"3"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&2b186594
&0&2808"
"4"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&3533f53f&
0&2008"
"5"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&3533f53f
&0&2808"
"6"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&16ea0f70
&0&2008"
"7"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&16ea0f70
&0&2808"
"8"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&1600be56
&0&2008"
"9"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&1600be56
&0&2808"
"10"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&8a8e882
&0&2008"
"11"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&8a8e882
&0&2808"
"12"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&6aa0d5f
&0&2008"
"13"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&6aa0d5f
&0&2808"
"14"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&3b7bf75
a&0&2008"
"15"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&3b7bf75
a&0&2808"
"16"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&285861a
a&0&2008"
"17"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&285861a
a&0&2808"
```

"18"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&27eba306&0&2008"
"19"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&27eba306&0&2808"
"20"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&3b940db4&0&2008"
"21"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&3b940db4&0&2808"
"22"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&33c2a78d&0&2008"
"23"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&33c2a78d&0&2808"
"24"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&26156fbc&0&2008"
"25"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&26156fbc&0&2808"
"26"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&1b9584b9&0&2008"
"27"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&1b9584b9&0&2808"
"28"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&20dd8e1e&0&2008"
"29"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&20dd8e1e&0&2808"
"30"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&14635f0&0&2008"
"31"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&14635f0&0&2808"
"32"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&13220302&0&2008"
"33"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&13220302&0&2808"
"34"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&1eb0d7f0&0&2008"
"35"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&1eb0d7f0&0&2808"
"36"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&fd960738&0&2008"
"37"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&fd960738&0&2808"
"38"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&466fdc18&0&2008"
"39"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&466fdc18&0&2808"
"40"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&f78fc9&0&2008"
"41"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&f78fc9&0&2808"
"42"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&36ded50f&0&2008"
"43"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&36ded50f&0&2808"
"44"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&273093ea&0&2008"
"45"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&273093ea&0&2808"
"46"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&301d439f&0&2008"
"47"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&301d439f&0&2808"
"48"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&348d8ddd&0&2008"
"49"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&348d8ddd&0&2808"
"50"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&214639d88&0&2008"
"51"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&214639d88&0&2808"
"52"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&1dded019&0&2008"
"53"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&1dded019&0&2808"
"54"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&13df1870&0&2008"
"55"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&13df1870&0&2808"
"56"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&fcf5f0&0&2008"
"57"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&fcf5f0&0&2808"
"58"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\\6&19905e4a&0&2008"
"59"="PCI\\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\\6&19905e4a&0&2808"

Qlogic 2350 VIA Card Settings

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000028
"Type"=dword:00000001
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,71,00,6c,00,32,00,33,00,30,00,30,\
00,2e,00,73,00,79,00,73,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters]
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters\Device]
"DriverParameters"="UseSameNN=0"
"MaximumSGList"=dword:00000021
"NumberOfRequests"=dword:00000096

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Parameters\PnpInterface]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2300\Enum]
"0"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&3633cca9&0&08"
"Count"=dword:00000010
"NextInstance"=dword:00000010
"1"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&251997c5&0&08"
"2"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&355c0d0f&0&08"
"3"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&8f8ade6&0&08"
"4"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&345102a3&0&08"
"5"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&370af391&0&08"
"6"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&266723a4&0&08"
"7"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&23ad32b6&0&08"
"8"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&7ab2207&0&08"
"9"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&a6512f5&0&08"
"10"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&279b1ea1&0&08"
"11"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&22da925c&0&08"
"12"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&331d07a6&0&08"
"13"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&b37b34f&0&08"
"14"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&3690080c&0&08"
"15"="PCI\\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\\5&3949f8fa&0&08"

C.1 Microsoft SQL Server 8.0 Configuration Parameters

name	minimum	maximum	config_value	run_value
Ad Hoc Distributed Queries	0	1	0	0
affinity I/O mask	-2147483648	2147483647	0	0
affinity mask	-2147483648	2147483647	-1	-1
affinity64 I/O mask	-2147483648	2147483647	0	0
affinity64 mask	-2147483648	2147483647	-1	-1
Agent XPs	0	1	0	0
allow updates	0	1	0	0
awe enabled	0	1	1	1
blocked process threshold	0	86400	0	0
c2 audit mode	0	1	0	0
clr enabled	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cross db ownership chaining	0	1	0	0
cursor threshold	-1	2147483647	-1	-1
Database Mail XPs	0	1	0	0
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
default trace enabled	0	1	0	0
disallow results from triggers	0	1	0	0
fill factor (%)	0	100	0	0
ft crawl bandwidth (max)	0	32767	100	100
ft crawl bandwidth (min)	0	32767	0	0
ft notify bandwidth (max)	0	32767	100	100
ft notify bandwidth (min)	0	32767	0	0
in-doubt xact resolution	0	2	0	0
index create memory (KB)	704	2147483647	0	0
lightweight pooling	0	1	1	1
locks	5000	2147483647	0	0
max degree of parallelism	0	64	1	1
max full-text crawl range	0	256	4	4
max server memory (MB)	16	2147483647	990000	990000
max text repl size (B)	0	2147483647	65536	65536
max worker threads	128	32767	2300	2300
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	0	8
nested triggers	0	1	1	1
network packet size (B)	512	32767	4096	4096
Ole Automation Procedures	0	1	0	0
PH timeout (s)	1	3600	60	60
precompute rank	0	2147483647	0	0
open objects	0	2147483647	0	0
priority boost	0	1	1	1
query governor cost limit	0	2147483647	0	0
query wait (s)	-1	2147483647	-1	-1
recovery interval (min)	0	32767	32767	32767
remote access	0	1	1	1
remote admin connections	0	1	0	0
remote login timeout (s)	0	2147483647	20	20
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	600	600
Replication XPs	0	1	0	0
RPC parameter data validation	0	1	0	0
scan for startup procs	0	1	0	0
server trigger recursion	0	1	1	1
set working set size	0	1	0	0
show advanced options	0	1	1	1
SMO and DMO XPs	0	1	1	1
SQL Mail XPs	0	1	0	0
transform noise words	0	1	0	0
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	0	0
User Instance Timeout	5	65535	10	10
user options	0	32767	0	0
Web Assistant Procedures	0	1	0	0
xp_cmdshell	0	1	0	0

C.2 Client System Configuration Parameters

Client Windows Server 2003 Settings

System Information report written at: 11/21/05 15:49:51
System Name: ORCAMASTER
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Standard Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	DL1
System Manufacturer	HP
System Model	ProLiant DL140
System Type	X86-based PC
Processor	x86 Family 15 Model 2 Stepping 9 GenuineIntel ~3065 Mhz
Processor	x86 Family 15 Model 2 Stepping 9 GenuineIntel ~3065 Mhz
BIOS Version/Date	Inventec 04403T03, 7/5/2004
SMBIOS Version	2.3
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)"
User Name	Not Available
Time Zone	Pacific Standard Time
Total Physical Memory	2,047.44 MB
Available Physical Memory	1.75 GB
Total Virtual Memory	3.85 GB
Available Virtual Memory	3.76 GB
Page File Space	2.00 GB
Page File	C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
I/O Port 0x00000000-0x000003AF	PCI bus
I/O Port 0x00000000-0x000003AF	Direct memory access controller
Memory Address 0xA0000-0xBFFFF	PCI bus
Memory Address 0xA0000-0xBFFFF	RAGE XL PCI Family (Microsoft Corporation)
I/O Port 0x000003B0-0x000003DF	PCI bus
I/O Port 0x000003B0-0x000003DF	RAGE XL PCI Family (Microsoft Corporation)

[DMA]

Resource	Device	Status
Channel 4	Direct memory access controller	OK

[Forced Hardware]

Device PNP Device ID

[I/O]

```
Resource    Device        Status
0x00000000-0x000003AF  PCI bus       OK
0x00000000-0x000003AF  Direct memory access controller          OK

0x000003B0-0x000003DF  PCI bus       OK
0x000003B0-0x000003DF  RAGE XL PCI Family (Microsoft Corporation)
            OK
0x000003E0-0x00000CF7  PCI bus       OK
0x00000D00-0x00000FFF  PCI bus       OK
0x0000D000-0x0000DFFF  PCI bus       OK
0x0000D800-0x0000D8FF  RAGE XL PCI Family (Microsoft Corporation)
            OK
0x000003C0-0x000003DF  RAGE XL PCI Family (Microsoft Corporation)
            OK
0x00000A79-0x00000A79  ISAPNP Read Data Port    OK
0x00000279-0x00000279  ISAPNP Read Data Port    OK
0x00000274-0x00000277  ISAPNP Read Data Port    OK
0x00000092-0x00000092  Motherboard resources    OK
0x000004D0-0x000004D1  Motherboard resources    OK
0x00000010-0x0000001F  Motherboard resources    OK
0x0000040B-0x0000040B  Motherboard resources    OK
0x000004D6-0x000004D6  Motherboard resources    OK
0x00000C06-0x00000C07  Motherboard resources    OK
0x00000C14-0x00000C14  Motherboard resources    OK
0x00000C49-0x00000C49  Motherboard resources    OK
0x00000C4A-0x00000C4A  Motherboard resources    OK
0x00000C50-0x00000C51  Motherboard resources    OK
0x00000C52-0x00000C52  Motherboard resources    OK
0x00000C6C-0x00000C6C  Motherboard resources    OK
0x00000C6F-0x00000C6F  Motherboard resources    OK
0x00000F50-0x00000F57  Motherboard resources    OK
0x00000C00-0x00000C01  Motherboard resources    OK
0x00000C98-0x00000C98  Motherboard resources    OK
0x00000CA3-0x00000CA3  Motherboard resources    OK
0x00000CD6-0x00000CD7  Motherboard resources    OK
0x0000002E-0x0000002F  Motherboard resources    OK
0x00000530-0x00000531  Motherboard resources    OK
0x00000400-0x0000041F  Motherboard resources    OK
0x00000440-0x0000045F  Motherboard resources    OK
0x00000480-0x0000049F  Motherboard resources    OK
0x000004C0-0x000004DF  Motherboard resources    OK
0x00000500-0x0000051F  Motherboard resources    OK
0x00000540-0x0000055F  Motherboard resources    OK
0x00000560-0x0000057F  Motherboard resources    OK
0x00000580-0x0000058F  Motherboard resources    OK
0x00000590-0x00000593  Motherboard resources    OK
0x00000020-0x00000021  Programmable interrupt controller        OK

0x000000A0-0x000000A1  Programmable interrupt controller        OK

0x00000080-0x0000008F  Direct memory access controller          OK
0x000000C0-0x000000DF  Direct memory access controller          OK

0x00000040-0x00000043  System timer OK
0x00000070-0x00000073  System CMOS/real time clock              OK

0x00000061-0x00000061  System speaker           OK
0x000000F0-0x000000FF  Numeric data processor   OK
0x00000060-0x00000060  Standard 101/102-Key or Microsoft Natural PS/2
Keyboard    OK
0x00000064-0x00000064  Standard 101/102-Key or Microsoft Natural PS/2
Keyboard    OK
0x0000DFF0-0x0000DFFF  Standard Dual Channel PCI IDE Controller
            OK
0x000001F0-0x000001F7  Primary IDE Channel      OK
0x000003F6-0x000003F6  Primary IDE Channel      OK
0x00000170-0x00000177  Secondary IDE Channel    OK
0x00000376-0x00000376  Secondary IDE Channel    OK
0x0000E000-0x0000EFFF  PCI bus       OK
0x0000E800-0x0000E8FF  QLogic QLA23xx PCI Fibre Channel Adapter
            OK

[IRQs]

Resource    Device        Status
IRQ 9       Microsoft ACPI-Compliant System       OK
IRQ 0       System timer OK
IRQ 8       System CMOS/real time clock           OK
IRQ 13      Numeric data processor    OK
IRQ 12      Microsoft PS/2 Mouse      OK
IRQ 1       Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
            OK
IRQ 14      Primary IDE Channel       OK
IRQ 10      Standard OpenHCD USB Host Controller OK
IRQ 22      QLogic QLA23xx PCI Fibre Channel Adapter      OK

IRQ 19      Broadcom NetXtreme Gigabit Ethernet #2        OK

IRQ 18      Broadcom NetXtreme Gigabit Ethernet   OK

[Memory]

Resource    Device        Status
0x0000-0x9FFFF          System board          OK
0xE0000-0xFFFFF         System board          OK
0x100000-0x7FFFFFFF     System board          OK
0xFFF80000-0xFFFFFFFF   System board          OK
0xA0000-0xBFFFF         PCI bus       OK
0xA0000-0xBFFFF         RAGE XL PCI Family (Microsoft Corporation)
            OK
0x80000000-0xFC5FFFFF   PCI bus       OK
0xFC600000-0xFC6FFFFF   PCI bus       OK
0xFC900000-0xFE9FFFFF   PCI bus       OK
0xFD000000-0xFDFFFFFF   RAGE XL PCI Family (Microsoft Corporation)
0xFE9FF000-0xFE9FFFFF   RAGE XL PCI Family (Microsoft Corporation)
            OK
0xFE9FE000-0xFE9FEFFF   Standard OpenHCD USB Host Controller OK

0xFC700000-0xFC7FFFFF   PCI bus       OK
0xFEA00000-0xFEAFFFFF   PCI bus       OK
0xFEAFF000-0xFEAFFFFF   QLogic QLA23xx PCI Fibre Channel Adapter
            OK
0xFC800000-0xFC8FFFFF   PCI bus       OK
0xFEB00000-0xFEBFFFFF   PCI bus       OK
0xFEBC0000-0xFEBCFFFF   Broadcom NetXtreme Gigabit Ethernet #2
            OK
0xFEBB0000-0xFEBBFFFF   Broadcom NetXtreme Gigabit Ethernet #2
            OK
0xFEBF0000-0xFEBFFFFF   Broadcom NetXtreme Gigabit Ethernet   OK

0xFEBE0000-0xFEBEFFFF   Broadcom NetXtreme Gigabit Ethernet   OK


[Components]



[Multimedia]



[Audio Codecs]

CODEC       Manufacturer  Description   Status        File          Version
            Size          Creation Date
c:\windows\system32\msg711.acm      Microsoft Corporation
            OK            C:\WINDOWS\system32\MSG711.ACM
            5.2.3790.0 (srv03_rtm.030324-2048)    10.00 KB (10,240
bytes)      4/12/2005 1:42 PM
c:\windows\system32\imaadp32.acm    Microsoft Corporation
            OK            C:\WINDOWS\system32\IMAADP32.ACM
            5.2.3790.0 (srv03_rtm.030324-2048)    15.50 KB (15,872
bytes)      4/12/2005 1:42 PM
c:\windows\system32\l3codeca.acm    Fraunhofer Institut Integrierte
Schaltungen IIS           Fraunhofer IIS MPEG Layer-3 Codec     OK
            C:\WINDOWS\system32\L3CODECA.ACM              1, 9, 0,
0305        284.00 KB (290,816 bytes)4/12/2005 1:43 PM
c:\windows\system32\msgsm32.acm     Microsoft Corporation
            OK            C:\WINDOWS\system32\MSGSM32.ACM
            5.2.3790.0 (srv03_rtm.030324-2048)    20.50 KB (20,992
bytes)      4/12/2005 1:42 PM
c:\windows\system32\msadp32.acm     Microsoft Corporation
            OK            C:\WINDOWS\system32\MSADP32.ACM
            5.2.3790.0 (srv03_rtm.030324-2048)    14.50 KB (14,848
bytes)      4/12/2005 1:42 PM
```

c:\windows\system32\tssoft32.acm DSP GROUP, INC.
OK C:\WINDOWS\system32\TSSOFT32.ACM1.01
9.50 KB (9,728 bytes) 4/12/2005 1:43 PM
c:\windows\system32\msg723.acm Microsoft Corporation
OK C:\WINDOWS\system32\MSG723.ACM
5.2.3790.1830 120.00 KB (122,880 bytes)
4/14/2005 10:01 AM
c:\windows\system32\msaud32.acm Microsoft Corporation Windows Media Audio Codec OK
C:\WINDOWS\system32\MSAUD32.ACM 8.00.00.4487288.00 KB (294,912 bytes) 4/12/2005 1:43 PM
c:\windows\system32\sl_anet.acm Sipro Lab Telecom Inc. Sipro Lab Telecom Audio Codec OK
C:\WINDOWS\system32\SL_ANET.ACM 3.02 84.00 KB (86,016 bytes) 4/12/2005 1:43 PM

[Video Codecs]

CODEC Manufacturer Description Status File Version
Size Creation Date
c:\windows\system32\msh263.drv Microsoft Corporation
OK C:\WINDOWS\system32\MSH263.DRV
5.2.3790.1830 288.00 KB (294,912 bytes)
4/14/2005 10:01 AM
c:\windows\system32\msh261.drv Microsoft Corporation
OK C:\WINDOWS\system32\MSH261.DRV
5.2.3790.1830 184.00 KB (188,416 bytes)
4/14/2005 10:01 AM
c:\windows\system32\tsbyuv.dll Microsoft Corporation
OK C:\WINDOWS\system32\TSBYUV.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB (8,192 bytes)
3/24/2003 5:50 PM
c:\windows\system32\msvidc32.dll Microsoft Corporation
OK C:\WINDOWS\system32\MSVIDC32.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 26.50 KB (27,136 bytes) 4/12/2005 1:42 PM
c:\windows\system32\msyuv.dll Microsoft Corporation
OK C:\WINDOWS\system32\MSYUV.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 16.50 KB (16,896 bytes) 3/24/2003 5:49 PM
c:\windows\system32\iyuv_32.dll Microsoft Corporation
OK C:\WINDOWS\system32\IYUV_32.DLL
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 46.50 KB (47,616 bytes) 4/14/2005 10:01 AM
c:\windows\system32\msrle32.dll Microsoft Corporation
OK C:\WINDOWS\system32\MSRLE32.DLL
5.2.3790.0 (srv03_rtm.030324-2048) 10.50 KB (10,752 bytes) 4/12/2005 1:42 PM

[CD-ROM]

Item Value

[Sound Device]

Item Value

[Display]

Item Value
Name RAGE XL PCI Family (Microsoft Corporation)
PNP Device ID
PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267A616A&0&18
Adapter TypeATI RAGE XL PCI (B41), ATI Technologies Inc. compatible

Adapter Description RAGE XL PCI Family (Microsoft Corporation)

Adapter RAM 8.00 MB (8,388,608 bytes)
Installed Drivers ati2drad.dll
Driver Version 5.10.3663.6013
INF File atiixpad.inf (ati2mpad section)
Color Planes 1
Color Table Entries 4294967296
Resolution 1024 x 768 x 85 hertz
Bits/Pixel 32
Memory Address 0xFD000000-0xFDFFFFFF
I/O Port 0x0000D800-0x0000D8FF
Memory Address 0xFE9FF000-0xFE9FFFFF
I/O Port 0x000003B0-0x000003DF
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver c:\windows\system32\drivers\ati2mpad.sys (5.10.3663.6013, 335.38 KB (343,424 bytes), 4/13/2005 3:31 AM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard

Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&39D9D85A&0
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver c:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 54.50 KB (55,808 bytes), 3/24/2003 5:01 PM)

[Pointing Device]

Item Value
Hardware Type Microsoft PS/2 Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F03\4&39D9D85A&0
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver c:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 54.50 KB (55,808 bytes), 3/24/2003 5:01 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000001] RAS Async Adapter
Adapter TypeNot Available
Product TypeRAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/18/2005 10:47 PM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter TypeNot Available
Product TypeWAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 11/18/2005 10:47 PM
Index 2

Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 66.00 KB (67,584 bytes), 4/12/2005 1:42 PM)

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 11/18/2005 10:47 PM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 61.00 KB (62,464 bytes), 4/12/2005 1:42 PM)

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 11/18/2005 10:47 PM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\raspppoe.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 40.00 KB (40,960 bytes), 4/12/2005 1:42 PM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 11/18/2005 10:47 PM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 19.50 KB (19,968 bytes), 4/12/2005 1:42 PM)

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 11/18/2005 10:47 PM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 91.00 KB (93,184 bytes), 4/12/2005 1:42 PM)

Name [00000007] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1648&SUBSYS_164814E4&REV_02\3&1070020&0&00
Last Reset 11/18/2005 10:47 PM
Index 7
Service Name b57w2k
IP Address 15.1.102.1
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0F:20:D0:A8:B3
Memory Address 0xFEBC0000-0xFEBCFFFF
Memory Address 0xFEBB0000-0xFEBBFFFF
IRQ Channel IRQ 19
Driver c:\windows\system32\drivers\b57xp32.sys (7.15.0.0 built by: WinDDK, 110.25 KB (112,896 bytes), 4/13/2005 4:24 PM)

Name [00000008] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet
Installed Yes
PNP Device ID PCI\VEN_14E4&DEV_1648&SUBSYS_164814E4&REV_02\3&1070020&0&01
Last Reset 11/18/2005 10:47 PM
Index 8
Service Name b57w2k
IP Address 192.168.100.1
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:0F:20:D0:A8:B4
Memory Address 0xFEBF0000-0xFEBFFFFF
Memory Address 0xFEBE0000-0xFEBEFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\b57xp32.sys (7.15.0.0 built by: WinDDK, 110.25 KB (112,896 bytes), 4/13/2005 4:24 PM)

Name [00000009] WAN Miniport (Network Monitor)
Adapter Type Not Available
Product Type WAN Miniport (Network Monitor)
Installed Yes
PNP Device ID ROOT\MS_NDISWANBH\0000
Last Reset 11/18/2005 10:47 PM
Index 9
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 91.00 KB (93,184 bytes), 4/12/2005 1:42 PM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP TCP Service Provider
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{356451BD-E9BA-4CDE-ADFA-95D45FC47A55}] SEQPACKET 3
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)
Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{356451BD-E9BA-4CDE-ADFA-95D45FC47A55}] DATAGRAM 3
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)
Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{6C8EF2B0-C570-41AD-BA17-47DDB867B581}] SEQPACKET 0
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)
Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{6C8EF2B0-C570-41AD-BA17-47DDB867B581}] DATAGRAM 0
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)
Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS [\Device\NetBT_Tcpip_{C09F1971-DD34-4CF7-86C3-79FC40F767E0}] SEQPACKET 1
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)
Message Oriented	Yes

Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{C09F1971-DD34-4CF7-86C3-79FC40F767E0}] DATAGRAM 1
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{F406FAC2-3DB8-4D61-8565-B2072EECB1F7}] SEQPACKET 2
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{F406FAC2-3DB8-4D61-8565-B2072EECB1F7}] DATAGRAM 2
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File c:\windows\system32\winsock.dll
Size 2.80 KB (2,864 bytes)
Version 3.10

File c:\windows\system32\wsock32.dll
Size 22.00 KB (22,528 bytes)
Version 5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item Value

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 74.53 GB (80,023,715,840 bytes)
Free Space 69.79 GB (74,938,527,744 bytes)
Volume Name
Volume Serial Number DCCFCBF1

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)
Model ST380011A
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 0
SCSI Target ID 0
Sectors/Track 63
Size 74.53 GB (80,023,749,120 bytes)
Total Cylinders 9,729
Total Sectors 156,296,385
Total Tracks 2,480,895
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 74.53 GB (80,023,716,864 bytes)
Partition Starting Offset 32,256 bytes

[SCSI]

Item Value
Name QLogic QLA23xx PCI Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&13C0B0C5&0&30
I/O Port 0x0000E800-0x0000E8FF
Memory Address 0xFEAFF000-0xFEAFFFFF
IRQ Channel IRQ 22
Driver c:\windows\system32\drivers\ql2300.sys (8.2.2.10 (W2K VI), 435.41 KB (445,858 bytes), 3/16/2005 8:45 AM)

[IDE]

Item Value
Name Standard Dual Channel PCI IDE Controller
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\VEN_1166&DEV_0213&SUBSYS_02121166&REV_A0\3&267A616A&0&79
I/O Port 0x0000DFF0-0x0000DFFF
Driver c:\windows\system32\drivers\pciide.sys (5.2.3790.0 (srv03_rtm.030324-2048), 5.50 KB (5,632 bytes), 3/24/2003 3:04 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&101988B2&0&0
I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/24/2003 3:04 PM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&101988B2&0&1
I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/24/2003 3:04 PM)

[Printing]

Name Driver Port Name Server Name

[Problem Devices]

Device PNP Device ID Error Code

[USB]

Device PNP Device ID
Standard OpenHCD USB Host Controller PCI\VEN_1166&DEV_0221&SUBSYS_02201166&REV_05\3&267A616A&0&7A

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State	Status	Error Control	Accept Pause	Accept Stop
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore	No	No
acpi	Microsoft ACPI Driver	c:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
adpu160m	adpu160m	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
adpu320	adpu320	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
afcnt	afcnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
afd	AFD Networking Support Environment	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes
aha154x	Aha154x	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
aic78u2	aic78u2	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
aic78xx	aic78xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
aliide	AliIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
alkernel	Altiris Kernel Driver	c:\windows\system32\drivers\alkernel.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes
asyncmac	RAS Asynchronous Media Driver	c:\windows\system32\drivers\asyncmac.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No
atapi	Standard IDE/ESDI Hard Disk Controller	c:\windows\system32\drivers\atapi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes
atdisk	Atdisk	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore	No	No
ati2mpad	ati2mpad	c:\windows\system32\drivers\ati2mpad.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	No	Yes
atmarpc	ATM ARP Client Protocol	c:\windows\system32\drivers\atmarpc.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No
audstub	Audio Stub Driver	c:\windows\system32\drivers\audstub.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes
b57w2k	Broadcom NetXtreme Gigabit Ethernet	c:\windows\system32\drivers\b57xp32.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No	Yes
beep	Beep	c:\windows\system32\drivers\beep.sys	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes
cbidf2k	cbidf2k	c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cd20xrnt	cd20xrnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cdfs	Cdfs	c:\windows\system32\drivers\cdfs.sys	File System Driver	No	Disabled	Stopped	OK	Normal	No	No
cdrom	Cdrom	c:\windows\system32\drivers\cdrom.sys	Kernel Driver	No	System	Stopped	OK	Ignore	No	No
changer	Changer	Not Available	Kernel Driver	No	System	Stopped	OK	Ignore	No	No
clusdisk	Cluster Disk Driver	c:\windows\system32\drivers\clusdisk.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cmdide	CmdIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cpqarray	Cpqarray	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cpqarry2	cpqarry2	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cpqcissm	cpqcissm	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
crcdisk	CRC Disk Filter Driver	c:\windows\system32\drivers\crcdisk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No	Yes
dac960nt	dac960nt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
dellcerc	dellcerc	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No
dfsdriver	DfsDriver	c:\windows\system32\drivers\dfs.sys	File System Driver	Yes	Boot	Running	OK	Normal	No	Yes

```
disk        Disk Driver   c:\windows\system32\drivers\disk.sys   Kernel
Driver      Yes           Boot          Running       OK          Normal
            No            Yes
dmboot      dmboot        c:\windows\system32\drivers\dmboot.sys
            Kernel Driver No            Disabled      Stopped     OK
            Normal        No            No
dmio        Logical Disk Manager Driver
            c:\windows\system32\drivers\dmio.sys   Kernel Driver Yes
            Boot          Running       OK          Normal      No
            Yes
dmload      dmload        c:\windows\system32\drivers\dmload.sys
            Kernel Driver Yes           Boot          Running     OK
            Normal        No            Yes
dpti2o      dpti2o        Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

fastfat     Fastfat       c:\windows\system32\drivers\fastfat.sys File
System Driver             No            Disabled      Stopped     OK
            Normal        No            No
fdc         Fdc           c:\windows\system32\drivers\fdc.sys     Kernel
Driver      No            System        Stopped       OK          Ignore
            No            No
fips        Fips          c:\windows\system32\drivers\fips.sys    Kernel
Driver      Yes           System        Running       OK          Normal
            No            Yes
flpydisk    Flpydisk      c:\windows\system32\drivers\flpydisk.sys
            Kernel Driver No            System        Stopped     OK
            Ignore        No            No
fltmgr      FltMgr        c:\windows\system32\drivers\fltmgr.sys  File
System Driver             Yes           Boot          Running     OK
            Normal        No            Yes
ftdisk      Volume Manager Driver
            c:\windows\system32\drivers\ftdisk.sys  Kernel Driver Yes
            Boot          Running       OK          Normal      No
            Yes
gpc         Generic Packet Classifier
            c:\windows\system32\drivers\msgpc.sys   Kernel Driver Yes
            Manual        Running       OK          Normal      No
            Yes
hpn         hpn           Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

hpt3xx      hpt3xx        Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

http        HTTP          c:\windows\system32\drivers\http.sys    Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
i2omgmt     i2omgmt       Not Available Kernel Driver No          System
            Stopped       OK            Normal        No          No

i2omp       i2omp         Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No
i8042prt    i8042 Keyboard and PS/2 Mouse Port Driver
            c:\windows\system32\drivers\i8042prt.sys    Kernel
Driver      Yes           System        Running       OK          Normal
            No            Yes
iirsp       iirsp         Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

intelide    IntelIde      Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

intelppm    Intel Processor Driver
            c:\windows\system32\drivers\intelppm.sys    Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
ip6fw       IPv6 Windows Firewall Driver
            c:\windows\system32\drivers\ip6fw.sys   Kernel Driver No
            Manual        Stopped       OK          Normal      No
            No
ipfilterdriver IP Traffic Filter Driver
            c:\windows\system32\drivers\ipfltdrv.sys    Kernel
Driver      No            Manual        Stopped       OK          Normal
            No            No
ipinip      IP in IP Tunnel Driver
            c:\windows\system32\drivers\ipinip.sys  Kernel Driver No
            Manual        Stopped       OK          Normal      No
            No
ipnat       IP Network Address Translator
            c:\windows\system32\drivers\ipnat.sys   Kernel Driver No
            Manual        Stopped       OK          Normal      No
            No
ipsec       IPSEC driver  c:\windows\system32\drivers\ipsec.sys  Kernel
Driver      Yes           System        Running       OK          Normal
            No            Yes
ipsraidn    ipsraidn      Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

irenum      IR Enumerator Service
            c:\windows\system32\drivers\irenum.sys      Kernel
Driver      No            Manual        Stopped       OK          Normal
            No            No
isapnp      PnP ISA/EISA Bus Driver
            c:\windows\system32\drivers\isapnp.sys  Kernel Driver Yes
            Boot          Running       OK          Critical    No
            Yes
kbdclass    Keyboard Class Driver
            c:\windows\system32\drivers\kbdclass.sys    Kernel
Driver      Yes           System        Running       OK          Normal
            No            Yes
ksecdd      KSecDD        c:\windows\system32\drivers\ksecdd.sys
            Kernel Driver Yes           Boot          Running     OK
            Normal        No            Yes

lp6nds35    lp6nds35      Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No
mnmdd       mnmdd         c:\windows\system32\drivers\mnmdd.sys
            Kernel Driver Yes           System        Running     OK
            Ignore        No            Yes
modem       Modem         c:\windows\system32\drivers\modem.sys
            Kernel Driver No            Manual        Stopped     OK
            Ignore        No            No
mouclass    Mouse Class Driver
            c:\windows\system32\drivers\mouclass.sys    Kernel
Driver      Yes           System        Running       OK          Normal
            No            Yes
mountmgr    Mount Point Manager
            c:\windows\system32\drivers\mountmgr.sys    Kernel
Driver      Yes           Boot          Running       OK          Normal
            No            Yes
mraid35x    mraid35x      Not Available Kernel Driver No          Disabled
            Stopped       OK            Normal        No          No

mrxdav      WebDav Client Redirector
            c:\windows\system32\drivers\mrxdav.sys      File
System Driver             No            Manual        Stopped     OK
            Normal        No            No
mrxsmb      MRXSMB        c:\windows\system32\drivers\mrxsmb.sys
            File System Driver          Yes           System      Running
            OK            Normal        No            Yes
msfs        Msfs          c:\windows\system32\drivers\msfs.sys    File
System Driver             Yes           System        Running     OK
            Normal        No            Yes
mssmbios    Microsoft System Management BIOS Driver
            c:\windows\system32\drivers\mssmbios.sys    Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
mup         Mup           c:\windows\system32\drivers\mup.sys     File
System Driver             Yes           Boot          Running     OK
            Normal        No            Yes
ndis        NDIS System Driver
            c:\windows\system32\drivers\ndis.sys    Kernel Driver Yes
            Boot          Running       OK          Normal      No
            Yes
ndistapi    Remote Access NDIS TAPI Driver
            c:\windows\system32\drivers\ndistapi.sys    Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
ndisuio     NDIS Usermode I/O Protocol
            c:\windows\system32\drivers\ndisuio.sys     Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
ndiswan     Remote Access NDIS WAN Driver
            c:\windows\system32\drivers\ndiswan.sys     Kernel
Driver      Yes           Manual        Running       OK          Normal
            No            Yes
```

```
ndproxy         NDIS Proxy          c:\windows\system32\drivers\ndproxy.sys
                Kernel Driver Yes   Manual      Running     OK
                Normal        No    Yes
netbios         NetBIOS Interface
                c:\windows\system32\drivers\netbios.sys                 File
System Driver               Yes   System      Running     OK
                Normal        No    Yes
netbt           NetBios over Tcpip
                c:\windows\system32\drivers\netbt.sys   Kernel Driver Yes
                System        Running     OK          Normal      No
                Yes
nfrd960         nfrd960             Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

nm              Network Monitor Driver
                c:\windows\system32\drivers\nmnt.sys    Kernel Driver No
                Manual        Stopped     OK          Normal      No
                No
npfs            Npfs          c:\windows\system32\drivers\npfs.sys      File
System Driver               Yes   System      Running     OK
                Normal        No    Yes
ntfs            Ntfs          c:\windows\system32\drivers\ntfs.sys      File
System Driver               Yes   Disabled    Running     OK
                Normal        No    Yes
null            Null          c:\windows\system32\drivers\null.sys      Kernel
Driver          Yes           System      Running     OK          Normal
                No            Yes
parport         Parport       c:\windows\system32\drivers\parport.sys
                Kernel Driver No    Manual      Stopped     OK
                Ignore        No    No
partmgr         Partition Manager
                c:\windows\system32\drivers\partmgr.sys                 Kernel
Driver          Yes           Boot        Running     OK          Normal
                No            Yes
pci             PCI Bus Driver              c:\windows\system32\drivers\pci.sys
                Kernel Driver Yes   Boot        Running     OK
                Critical      No    Yes
pciide          PCIIde        c:\windows\system32\drivers\pciide.sys    Kernel
Driver          Yes           Boot        Running     OK          Normal
                No            Yes
pcmcia          Pcmcia        c:\windows\system32\drivers\pcmcia.sys
                Kernel Driver No    Disabled    Stopped     OK
                Normal        No    No
pdcomp          PDCOMP              Not Available Kernel Driver No          Manual
                Stopped       OK    Ignore      No          No

pdframe         PDFRAME             Not Available Kernel Driver No          Manual
                Stopped       OK    Ignore      No          No

pdreli          PDRELI              Not Available Kernel Driver No          Manual
                Stopped       OK    Ignore      No          No

pdrframe        PDRFRAME            Not Available Kernel Driver No          Manual
                Stopped       OK    Ignore      No          No
perc2           perc2               Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

perc2hib        perc2hib            Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

pgtrackr01      Page Tracker1 for X86perfsys
                \??\c:\lop1.1\pgtrackr01.sys            Kernel Driver No
                Manual        Stopped     OK          Ignore      No
                No
pptpminiport    WAN Miniport (PPTP)
                c:\windows\system32\drivers\raspptp.sys                 Kernel
Driver          Yes           Manual      Running     OK          Normal
                No            Yes
processor       Processor Driver
                c:\windows\system32\drivers\processr.sys                Kernel
Driver          No            Manual      Stopped     OK          Normal
                No            No
ptilink         Direct Parallel Link Driver
                c:\windows\system32\drivers\ptilink.sys Kernel Driver Yes
                Manual        Running     OK          Normal      No
                Yes
ql1080          ql1080              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql10wnt         Ql10wnt             Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql12160         ql12160             Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql1240          ql1240              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql1280          ql1280              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql2100          ql2100              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql2200          ql2200              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

ql2300          ql2300              c:\windows\system32\drivers\ql2300.sys Kernel
Driver          Yes           Boot        Running     OK          Normal
                No            Yes
qlvika          qlvika              c:\windows\system32\drivers\qlvika.sys Kernel
Driver          Yes           Auto        Running     OK          Normal
                No            Yes

rasacd          Remote Access Auto Connection Driver
                c:\windows\system32\drivers\rasacd.sys  Kernel Driver Yes
                System        Running     OK          Normal      No
rasl2tp         WAN Miniport (L2TP)
                c:\windows\system32\drivers\rasl2tp.sys Kernel Driver Yes
                Manual        Running     OK          Normal      No
                Yes
raspppoe        Remote Access PPPOE Driver
                c:\windows\system32\drivers\raspppoe.sys                Kernel
Driver          Yes           Manual      Running     OK          Normal
                No            Yes
raspti          Direct Parallel
                c:\windows\system32\drivers\raspti.sys  Kernel Driver Yes
                Manual        Running     OK          Normal      No
                Yes
rdbss           Rdbss         c:\windows\system32\drivers\rdbss.sys     File
System Driver               Yes   System      Running     OK
                Normal        No    Yes
rdpcdd          RDPCDD        c:\windows\system32\drivers\rdpcdd.sys
                Kernel Driver Yes   System      Running     OK
                Ignore        No    Yes
rdpdr           Terminal Server Device Redirector Driver
                c:\windows\system32\drivers\rdpdr.sys   Kernel Driver Yes
                Manual        Running     OK          Normal      No
                Yes
rdpwd           RDPWD         c:\windows\system32\drivers\rdpwd.sys     Kernel
Driver          Yes           Manual      Running     OK          Ignore
                No            Yes
secdrv          Secdrv        c:\windows\system32\drivers\secdrv.sys    Kernel
Driver          No            Manual      Stopped     OK          Normal
                No            No
serenum         Serenum Filter Driver
                c:\windows\system32\drivers\serenum.sys                 Kernel
Driver          No            Manual      Stopped     OK          Normal
                No            No
serial          Serial port driver
                c:\windows\system32\drivers\serial.sys  Kernel Driver No
                System        Stopped     OK          Ignore      No
                No
sfloppy         Sfloppy       c:\windows\system32\drivers\sfloppy.sys
                Kernel Driver No    System      Stopped     OK
                Ignore        No    No
simbad          Simbad              Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

sparrow         Sparrow             Not Available Kernel Driver No          Disabled
                Stopped       OK    Normal      No          No

srv             Srv           c:\windows\system32\drivers\srv.sys       File
System Driver               Yes   Manual      Running     OK
                Normal        No    Yes
```

```
swenum          Software Bus Driver
                c:\windows\system32\drivers\swenum.sys                  Kernel
Driver          Yes             Manual          Running         OK      Normal
                No              Yes
symc810         symc810         Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

symc8xx         symc8xx         Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

symmpi          symmpi          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

sym_hi          sym_hi          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

sym_u3          sym_u3          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

tcpip           TCP/IP Protocol Driver
                c:\windows\system32\drivers\tcpip.sys   Kernel Driver   Yes
                System          Running         OK              Normal  No
                Yes
tdpipe          TDPIPE          c:\windows\system32\drivers\tdpipe.sys  Kernel
Driver          No              Manual          Stopped         OK      Ignore
                No              No
tdtcp           TDTCP           c:\windows\system32\drivers\tdtcp.sys   Kernel
Driver          Yes             Manual          Running         OK      Ignore
                No              Yes
termdd          Terminal Device Driver
                c:\windows\system32\drivers\termdd.sys                  Kernel
Driver          Yes             System          Running         OK      Normal
                No              Yes
toside          TosIde          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

udfs            Udfs            c:\windows\system32\drivers\udfs.sys    File
System Driver                   No              Disabled        Stopped OK
                Normal          No              No
ultra           ultra           Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

update          Microcode Update Driver
                c:\windows\system32\drivers\update.sys                  Kernel
Driver          Yes             Manual          Running         OK      Normal
                No              Yes
usbhub          USB2 Enabled Hub
                c:\windows\system32\drivers\usbhub.sys                  Kernel
Driver          Yes             Manual          Running         OK      Normal
                No              Yes

usbohci         Microsoft USB Open Host Controller Miniport Driver
                c:\windows\system32\drivers\usbohci.sys                 Kernel
Driver          Yes             Manual          Running         OK      Normal
usbstor         USB Mass Storage Driver
                c:\windows\system32\drivers\usbstor.sys                 Kernel
Driver          No              Manual          Stopped         OK      Normal
                No              No
vgasave         VGA Display Controller.
                c:\windows\system32\drivers\vga.sys     Kernel Driver   Yes
                System          Running         OK              Ignore  No
                Yes
viaide          ViaIde          Not Available   Kernel Driver   No      Disabled
                Stopped         OK              Normal          No      No

volsnap         Storage volumes
                c:\windows\system32\drivers\volsnap.sys                 Kernel
Driver          Yes             Boot            Running         OK      Normal
                No              Yes
wanarp          Remote Access IP ARP Driver
                c:\windows\system32\drivers\wanarp.sys                  Kernel
Driver          Yes             Manual          Running         OK      Normal
                No              Yes
wdica           WDICA           Not Available   Kernel Driver   No      Manual
                Stopped         OK              Ignore          No      No

wlbs            Network Load Balancing
                c:\windows\system32\drivers\wlbs.sys    Kernel Driver   No
                Manual          Stopped         OK              Normal  No
                No
x86perfsys      Low Overhead Profiler   \??\c:\lop1.1\x86perfsys.sys
                Kernel Driver   No              Manual          Stopped OK
                Ignore          No              No

[Signed Drivers]

Device Name     Signed          Device Class    Driver Version          Driver
Date            Manufacturer    INF Name        Driver Name     Device ID
Microsoft System Management BIOS Driver         Yes             SYSTEM
                5.2.3790.1830   10/1/2002       (Standard system
devices)        machine.inf     Not Available   ROOT\SYSTEM\0002
Microcode Update Device         Yes             SYSTEM          5.2.3790.0
                10/1/2002       (Standard system devices)       machine.inf     Not
Available       ROOT\SYSTEM\0001
Plug and Play Software Device Enumerator        Yes             SYSTEM
                5.2.3790.0      10/1/2002       (Standard system devices)
                machine.inf     Not Available   ROOT\SYSTEM\0000
Terminal Server Mouse Driver    Yes             SYSTEM
                5.2.3790.0      10/1/2002       (Standard system devices)
                machine.inf     Not Available   ROOT\RDP_MOU\0000
Terminal Server Keyboard Driver Yes             SYSTEM
                5.2.3790.0      10/1/2002       (Standard system devices)
                machine.inf     Not Available   ROOT\RDP_KBD\0000
Terminal Server Device Redirector       Yes             SYSTEM
                5.2.3790.0      10/1/2002       (Standard system devices)
Direct Parallel machine.inf     Not Available   NET             5.2.3790.0
                10/1/2002       Microsoft       netrasa.inf     Not Available
                ROOT\MS_PTIMINIPORT\0000
WAN Miniport (PPTP)             Yes             NET             5.2.3790.0
                10/1/2002       Microsoft       netrasa.inf     Not Available
                ROOT\MS_PPTPMINIPORT\0000
WAN Miniport (PPPOE)            Yes             NET             5.2.3790.0
                10/1/2002       Microsoft       netrasa.inf     Not Available
                ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (IP)               Yes             NET             5.2.3790.0
                10/1/2002       Microsoft       netrasa.inf     Not Available
                ROOT\MS_NDISWANIP\0000
WAN Miniport (Network Monitor)  Yes             NET
                5.2.3790.1830   10/1/2002       Microsoft
                netrasa.inf     Not Available   ROOT\MS_NDISWANBH\0000

WAN Miniport (L2TP)             Yes             NET             5.2.3790.0
                10/1/2002       Microsoft       netrasa.inf     Not Available
                ROOT\MS_L2TPMINIPORT\0000
Video Codecs    Yes             MEDIA           5.2.3790.0      10/1/2002       (Standard
system devices)                 wave.inf        Not Available
                ROOT\MEDIA\MS_MMVID
Legacy Video Capture Devices    Yes             MEDIA
                5.2.3790.0      10/1/2002       (Standard system devices)       wave.inf
                Not Available   ROOT\MEDIA\MS_MMVCD
Media Control Devices           Yes             MEDIA           5.2.3790.0
                10/1/2002       (Standard system devices)       wave.inf        Not
Available       ROOT\MEDIA\MS_MMMCI
Legacy Audio Drivers            Yes             MEDIA           5.2.3790.0
                10/1/2002       (Standard system devices)       wave.inf        Not
Available       ROOT\MEDIA\MS_MMDRV
Audio Codecs    Yes             MEDIA           5.2.3790.0      10/1/2002       (Standard
system devices)                 wave.inf        Not Available
                ROOT\MEDIA\MS_MMACM
Low Overhead Profiler           Not Available   LEGACYDRIVER            Not
Available       Not Available   Not Available   Not Available   Not Available
                ROOT\LEGACY_X86PERFSYS\0000
Remote Access IP ARP Driver     Not Available   LEGACYDRIVER
                Not Available   Not Available   Not Available   Not
Available       ROOT\LEGACY_WANARP\0000
volsnap         Not Available   LEGACYDRIVER            Not Available   Not
Available       Not Available   Not Available   Not Available
                ROOT\LEGACY_VOLSNAP\0000
VGA Display Controller.         Not Available   LEGACYDRIVER            Not
Available       Not Available   Not Available   Not Available   Not Available
                ROOT\LEGACY_VGASAVE\0000
TDTCP           Not Available   LEGACYDRIVER            Not Available   Not
Available       Not Available   Not Available   Not Available
                ROOT\LEGACY_TDTCP\0000
```

TCP/IP Protocol Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_TCPIP\0000
sacdrv Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_SACDRV\0000
RDPWD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPWD\0000
RDPCDD Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RDPCDD\0000
Remote Access Auto Connection Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_RASACD\0000
qlvika Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_QLVIKA\0000
Page Tracker1 for X86perfsys Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_PGTRACKR01\0000
Partition Manager Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_PARTMGR\0000
Null Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NULL\0000
Network Monitor Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NM\0000
NetBios over Tcpip Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NETBT\0000
NDProxy Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDPROXY\0000
NDIS Usermode I/O Protocol Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISUIO\0000
Remote Access NDIS TAPI Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDISTAPI\0000
NDIS System Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_NDIS\0000
mountmgr Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_MOUNTMGR\0000
modem Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_MODEM\0000

mnmdd Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_MNMDD\0000
ksecdd Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_KSECDD\0000
IPSEC driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_IPSEC\0000
HTTP Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_HTTP\0000
Generic Packet Classifier Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_GPC\0000
Fips Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_FIPS\0000
dmload Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_DMLOAD\0000
dmboot Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_DMBOOT\0000
CRC Disk Filter Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_CRCDISK\0000
Beep Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_BEEP\0000
Altiris Kernel Driver Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_ALKERNEL\0000
AFD Networking Support Environment Not Available LEGACYDRIVER Not Available Not Available Not Available Not Available Not Available ROOT\LEGACY_AFD\0000
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATUREB107B107OFFSET7E00LENGTH12A1C90400
Volume Manager Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\FTDISK\0000
Logical Disk Manager Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices) machine.inf Not Available ROOT\DMIO\0000
ACPI Fixed Feature Button Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\FIXEDBUTTON\2&DABA3FF&0
Broadcom NetXtreme Gigabit Ethernet Yes NET 7.15.0.0 8/29/2003 Broadcom oem0.inf Not Available PCI\VEN_14E4&DEV_1648&SUBSYS_164814E4&REV_02\3&1070020&0&01

Broadcom NetXtreme Gigabit Ethernet Yes NET 7.15.0.0 8/29/2003 Broadcom oem0.inf Not Available PCI\VEN_14E4&DEV_1648&SUBSYS_164814E4&REV_02\3&1070020&0&00
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\2

QLOGIC PSEUDO LUN Yes SYSTEM 8.2.2.10 4/16/2003 QLogic Corp oem2.inf Not Available SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_LUN&REV_\4&13636C63&0&07F0
QLogic QLA23xx PCI Fibre Channel Adapter Yes SCSIADAPTER 8.2.2.10 4/16/2003 QLogic oem1.inf Not Available PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\3&13C0B0C5&0&30
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0A03\1

ServerWorks Grand Champion CIOB_E - I/O Bridge with GB Ethernet Yes SYSTEM 5.2.3790.0 10/1/2002 ServerWorks (RCC) machine.inf Not Available PCI\VEN_1166&DEV_0110&SUBSYS_00000000&REV_12\3&267A616A&0&82
ServerWorks Grand Champion CIOB_E - I/O Bridge with GB Ethernet Yes SYSTEM 5.2.3790.0 10/1/2002 ServerWorks (RCC) machine.inf Not Available PCI\VEN_1166&DEV_0110&SUBSYS_00000000&REV_12\3&267A616A&0&80
Serverworks Champion CSB6 - SouthBridge 6 LPC Yes SYSTEM 5.2.3790.0 10/1/2002 ServerWorks (RCC) machine.inf Not Available PCI\VEN_1166&DEV_0227&SUBSYS_00000000&REV_00\3&267A616A&0&7B
USB Root Hub Yes USB 5.2.3790.0 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available USB\ROOT_HUB\4&331E2EA4&0

Standard OpenHCD USB Host Controller Yes USB 5.2.3790.0 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available PCI\VEN_1166&DEV_0221&SUBSYS_02201166&REV_05\3&267A616A&0&7A
Secondary IDE Channel Yes HDC 5.2.3790.0 10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available PCIIDE\IDECHANNEL\4&101988B2&0&1
Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard disk drives) disk.inf Not Available IDE\DISKST380011A_____8.01____\4A3544564745343920202020202020202020
Primary IDE Channel Yes HDC 5.2.3790.0 10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available PCIIDE\IDECHANNEL\4&101988B2&0&0

Standard Dual Channel PCI IDE Controller Yes HDC
5.2.3790.0 10/1/2002 (Standard IDE ATA/ATAPI
controllers) mshdc.inf Not Available
PCI\VEN_1166&DEV_0213&SUBSYS_02121166&REV_A0\3&267
A616A&0&79
Standard 101/102-Key or Microsoft Natural PS/2 Keyboard Yes
KEYBOARD 5.2.3790.0 10/1/2002 (Standard keyboards)
keyboard.inf Not Available ACPI\PNP0303\4&39D9D85A&0

Microsoft PS/2 Mouse Yes MOUSE 5.2.3790.0
10/1/2002 Microsoft msmouse.inf Not Available
ACPI\PNP0F03\4&39D9D85A&0
Numeric data processor Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\PNP0C04\4&39D9D85A&0
System speaker Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\PNP0800\4&39D9D85A&0
System CMOS/real time clock Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard system devices)
machine.inf Not Available ACPI\PNP0B00\4&39D9D85A&0

System timer Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available
ACPI\PNP0100\4&39D9D85A&0
Direct memory access controller Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard system devices)
machine.inf Not Available ACPI\PNP0200\4&39D9D85A&0

Programmable interrupt controller Yes SYSTEM
5.2.3790.0 10/1/2002 (Standard system devices)
machine.inf Not Available ACPI\PNP0000\4&39D9D85A&0

Motherboard resources Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\PNP0C02\4&39D9D85A&0
ISAPNP Read Data Port Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available ISAPNP\READDATAPORT\0
PCI standard ISA bridge Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available
PCI\VEN_1166&DEV_0203&SUBSYS_00000008&REV_A0\3&267
A616A&0&78
Plug and Play Monitor Yes MONITOR 5.1.2001.0 6/6/2001
(Standard monitor types) monitor.inf Not Available
DISPLAY\AVO0402\4&89B5141&0&80000001&00&03
RAGE XL PCI Family (Microsoft Corporation) Yes DISPLAY
5.10.2600.6014 8/8/2001 ATI Technologies Inc.
atiixpad.inf Not Available
PCI\VEN_1002&DEV_4752&SUBSYS_001E0E11&REV_27\3&267
A616A&0&18

ServerWorks Grand Champion CMIC_SL - NorthBridge Super Lite Yes
SYSTEM 5.2.3790.0 10/1/2002 ServerWorks (RCC)
machine.inf Not Available
PCI\VEN_1166&DEV_0017&SUBSYS_00000008&REV_00\3&267
A616A&0&00
ServerWorks Grand Champion CMIC_SL - NorthBridge Super Lite Yes
SYSTEM 5.2.3790.0 10/1/2002 ServerWorks (RCC)
machine.inf Not Available
PCI\VEN_1166&DEV_0017&SUBSYS_00000008&REV_32\3&267
A616A&0&00
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\PNP0A03\0

System board Yes SYSTEM 5.2.3790.0
10/1/2002 (Standard system devices) machine.inf Not
Available ACPI\PNP0C01\2&DABA3FF&0
Intel Processor Yes PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-_X86_FAMILY_15_MODEL_2_2

Intel Processor Yes PROCESSOR 5.2.3790.1830
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-_X86_FAMILY_15_MODEL_2_1

Microsoft ACPI-Compliant System Yes SYSTEM
5.2.3790.0 10/1/2002 Microsoft acpi.inf Not
Available ACPI_HAL\PNP0C08\0
ACPI Multiprocessor PC Yes COMPUTER 5.2.3790.0
10/1/2002 (Standard computers) hal.inf Not
Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not Available Not Available Not
Available Not Available Not Available HTREE\ROOT\0

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\Sy
stem32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN
<SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 2 Stepping 9, GenuineIntel
<SYSTEM>
PROCESSOR_REVISION 0209 <SYSTEM>
NUMBER_OF_PROCESSORS 2 <SYSTEM>
ClusterLog C:\WINDOWS\Cluster\cluster.log <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
<SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>

TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL
SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL
SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp DL1\Administrator

TMP %USERPROFILE%\Local Settings\Temp DL1\Administrator

[Print Jobs]

Document Size Owner Notify Status Time
Submitted Start Time Until Time Elapsed TimePages Printed
Job ID Priority Parameters Driver Print
Processor Host Print Queue Data Type Name

[Network Connections]

Local Name Remote Name Type Status User
Name

[Running Tasks]

Name Path Process ID Priority Min Working Set
Max Working Set Start Time Version Size
File Date
system idle process Not Available 0 0 Not
Available Not Available Not Available Not Available Not Available Not
Available
system Not Available 4 8 0 1413120
Not Available Not Available Not Available Not Available
smss.exe Not Available 556 11 204800 1413120
11/18/2005 10:47 PM Not Available Not Available Not
Available
csrss.exe c:\windows\system32\csrss.exe 620 13
204800 1413120 11/18/2005 10:47 PM
5.2.3790.0 (srv03_rtm.030324-2048) 4.00 KB (4,096 bytes)
4/12/2005 1:42 PM
winlogon.exe c:\windows\system32\winlogon.exe 660 13
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.00
KB (508,928 bytes) 4/14/2005 10:00 AM

services.exe c:\windows\system32\services.exe 704 9
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 107.50
KB (110,080 bytes) 4/12/2005 1:43 PM
lsass.exe c:\windows\system32\lsass.exe 716 9
204800 1413120 11/18/2005 10:47 PM
5.2.3790.0 (srv03_rtm.030324-2048) 13.00 KB (13,312
bytes) 4/12/2005 1:42 PM
svchost.exe c:\windows\system32\svchost.exe 932 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe 1020 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe 1112 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe 1192 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe 1204 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
spoolsv.exe c:\windows\system32\spoolsv.exe 1724 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 57.00 KB
(58,368 bytes) 4/14/2005 10:01 AM
msdtc.exe c:\windows\system32\msdtc.exe 1756 8
204800 1413120 11/18/2005 10:47 PM
2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 6.00 KB
(6,144 bytes)4/14/2005 10:01 AM
aclient.exe c:\program files\altiris\aclient\aclient.exe 1916
8 204800 1413120 11/18/2005 10:47 PM
6.1.375 4.62 MB (4,845,644 bytes) 5/3/2005 8:59 AM

svchost.exe c:\windows\system32\svchost.exe 1960 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
svchost.exe c:\windows\system32\svchost.exe 352 8
204800 1413120 11/18/2005 10:47 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
rsys.exe c:\program files\benchcraft\rsys.exe 420 8
204800 1413120 11/18/2005 10:47 PM Not
Available 9.50 KB (9,728 bytes) 9/25/2003 6:41 PM

svchost.exe c:\windows\system32\svchost.exe 1472 8
204800 1413120 11/18/2005 10:48 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
wmiprvse.exe c:\windows\system32\wbem\wmiprvse.exe
1440 8 204800 1413120
11/18/2005 10:49 PM 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 203.00 KB (207,872 bytes)
4/14/2005 10:01 AM
logon.scr c:\windows\system32\logon.scr 496 4
204800 1413120 11/18/2005 10:57 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50
KB (509,440 bytes) 4/14/2005 10:01 AM
dllhost.exe c:\windows\system32\dllhost.exe 1772 8
204800 1413120 11/19/2005 9:17 AM
5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes)
4/12/2005 1:42 PM
inetinfo.exe c:\windows\system32\inetsrv\inetinfo.exe 2320
8 204800 1413120 11/19/2005 10:53 AM
6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:02 AM
svchost.exe c:\windows\system32\svchost.exe 260 8
204800 1413120 11/19/2005 10:53 AM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB
(14,336 bytes) 4/14/2005 10:01 AM
wmiprvse.exe c:\windows\system32\wbem\wmiprvse.exe
3604 8 204800 1413120
11/21/2005 3:48 PM 5.2.3790.1830
(srv03_sp1_rtm.050324-1447) 203.00 KB (207,872 bytes)
4/14/2005 10:01 AM

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
csrss	5.2.3790.0 (srv03_rtm.030324-2048)	4.00 KB (4,096 bytes)	4/12/2005 1:42 PM	Microsoft Corporation	c:\windows\system32\csrss.exe
ntdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	748.50 KB (766,464 bytes)	4/12/2005 1:42 PM	Microsoft Corporation	c:\windows\system32\ntdll.dll
csrsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	33.00 KB (33,792 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\csrsrv.dll
basesrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	51.50 KB (52,736 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\basesrv.dll
winsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	290.50 KB (297,472 bytes)	4/14/2005 10:00 AM	Microsoft Corporation	c:\windows\system32\winsrv.dll
gdi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	273.00 KB (279,552 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\gdi32.dll
advapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	605.50 KB (620,032 bytes)	4/12/2005 1:42 PM	Microsoft Corporation	c:\windows\system32\advapi32.dll
kernel32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,014.00 KB (1,038,336 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\kernel32.dll
rpcrt4	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	627.00 KB (642,048 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\rpcrt4.dll
user32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	574.50 KB (588,288 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\user32.dll
sxs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	743.50 KB (761,344 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\sxs.dll
winlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	497.00 KB (508,928 bytes)	4/14/2005 10:00 AM	Microsoft Corporation	c:\windows\system32\winlogon.exe
crypt32	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	582.00 KB (595,968 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\crypt32.dll
msasn1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	56.50 KB (57,856 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\msasn1.dll
msvcrt	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)	340.50 KB (348,672 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\msvcrt.dll
nddeapi	5.2.3790.0 (srv03_rtm.030324-2048)	16.00 KB (16,384 bytes)	4/12/2005 1:42 PM	Microsoft Corporation	c:\windows\system32\nddeapi.dll
profmap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	22.50 KB (23,040 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\profmap.dll
netapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	341.50 KB (349,696 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\netapi32.dll
userenv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	771.00 KB (789,504 bytes)	4/12/2005 1:43 PM	Microsoft Corporation	c:\windows\system32\userenv.dll
psapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	20.00 KB (20,480 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\psapi.dll
regapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	55.00 KB (56,320 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\regapi.dll
secur32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	64.00 KB (65,536 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\secur32.dll
setupapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.03 MB (1,079,808 bytes)	4/12/2005 1:43 PM	Microsoft Corporation	c:\windows\system32\setupapi.dll
version	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	18.00 KB (18,432 bytes)	4/14/2005 10:01 AM	Microsoft Corporation	c:\windows\system32\version.dll

winsta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 54.50 KB (55,808 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\winsta.dll
ws2_32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 82.00 KB (83,968 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\ws2_32.dll
ws2help 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.50 KB (19,968 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\ws2help.dll
msgina 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.16 MB (1,211,904 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msgina.dll
shsvcs 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 131.50 KB (134,656 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\shsvcs.dll
shlwapi 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 313.50 KB (321,024 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\shlwapi.dll
sfc 5.2.3790.0 (srv03_rtm.030324-2048) 4.50 KB (4,608 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\sfc.dll
sfc_os 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 138.00 KB (141,312 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\sfc_os.dll
wintrust 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 162.00 KB (165,888 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wintrust.dll
imagehlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 145.50 KB (148,992 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\imagehlp.dll
ole32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.19 MB (1,245,184 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ole32.dll
comctl32 6.0 (srv03_sp1_rtm.050324-1447) 1.00 MB (1,051,136 bytes) 3/24/2005 8:41 PM Microsoft Corporation c:\windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.3790.1830_x-ww_7ae38ccf\comctl32.dll

winscard 5.2.3790.0 (srv03_rtm.030324-2048) 98.50 KB (100,864 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\winscard.dll
wtsapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.00 KB (19,456 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wtsapi32.dll
winmm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 172.50 KB (176,640 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\winmm.dll
shell32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 7.99 MB (8,379,392 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\shell32.dll

wldap32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 174.50 KB (178,688 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wldap32.dll
rsaenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 183.98 KB (188,392 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rsaenh.dll
cscdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 100.00 KB (102,400 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cscdll.dll
dimsntfy 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.00 KB (19,456 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\dimsntfy.dll
wlnotify 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 94.50 KB (96,768 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wlnotify.dll
mpr 5.2.3790.0 (srv03_rtm.030324-2048) 56.00 KB (57,344 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\mpr.dll
oleaut32 5.2.3790.1830 543.00 KB (556,032 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\oleaut32.dll
winspool 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 147.00 KB (150,528 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\winspool.drv
comctl32 5.82 (srv03_sp1_rtm.050324-1447) 585.00 KB (599,040 bytes) 3/24/2005 8:41 PM Microsoft Corporation c:\windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_5.82.3790.1830_x-ww_1b6f474a\comctl32.dll

uxtheme 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 202.00 KB (206,848 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\uxtheme.dll
mprapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 89.00 KB (91,136 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mprapi.dll
activeds 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 194.00 KB (198,656 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\activeds.dll
adsldpc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 146.00 KB (149,504 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\adsldpc.dll
credui 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 162.00 KB (165,888 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\credui.dll
atl 3.05.2283 83.00 KB (84,992 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\atl.dll

rtutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 34.50 KB (35,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rtutils.dll
samlib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 46.50 KB (47,616 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\samlib.dll

clbcatq 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 502.50 KB (514,560 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\clbcatq.dll
comres 2001.12.4720.0 (srv03_rtm.030324-2048) 778.00 KB (796,672 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\comres.dll
wbemprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 20.50 KB (20,992 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbemprox.dll
wbemcomn 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 221.00 KB (226,304 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbemcomn.dll
xpsp2res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.76 MB (2,897,920 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\xpsp2res.dll
wbemsvc 5.2.3790.0 (srv03_rtm.030324-2048) 42.50 KB (43,520 bytes) 4/13/2005 3:12 PM Microsoft Corporation c:\windows\system32\wbem\wbemsvc.dll
fastprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 471.00 KB (482,304 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll
msvcp60 6.05.2144.0 388.00 KB (397,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msvcp60.dll

ntdsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 71.00 KB (72,704 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll
dnsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 153.50 KB (157,184 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\dnsapi.dll
services 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 107.50 KB (110,080 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\services.exe
ncobjapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.00 KB (36,864 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll
scesrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 327.00 KB (334,848 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\scesrv.dll
authz 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 66.50 KB (68,096 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\authz.dll
umpnpmgr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 126.50 KB (129,536 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\umpnpmgr.dll
eventlog 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 67.50 KB (69,120 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\eventlog.dll
lsass 5.2.3790.0 (srv03_rtm.030324-2048) 13.00 KB (13,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\lsass.exe
lsasrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 803.00 KB (822,272 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\lsasrv.dll

samsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 450.50 KB (461,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\samsrv.dll
cryptdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 32.00 KB (32,768 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptdll.dll
msprivs 5.2.3790.0 (srv03_rtm.030324-2048) 46.50 KB (47,616 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msprivs.dll
kerberos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 340.50 KB (348,672 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\kerberos.dll
msv1_0 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 141.00 KB (144,384 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msv1_0.dll
iphlpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 92.50 KB (94,720 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll
netlogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 409.50 KB (419,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netlogon.dll
w32time 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 222.00 KB (227,328 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\w32time.dll
schannel 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 141.00 KB (144,384 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\schannel.dll
wdigest 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 74.00 KB (75,776 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wdigest.dll
rassfm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 23.00 KB (23,552 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rassfm.dll
kdcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 213.50 KB (218,624 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.45 MB (1,516,032 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsa.dll
esent 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1,022.50 KB (1,047,040 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\esent.dll
ntdsatq 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 29.50 KB (30,208 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntdsatq.dll
mswsock 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 250.50 KB (256,512 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mswsock.dll
scecli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 186.50 KB (190,976 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\scecli.dll
ws03res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 793.50 KB (812,544 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\ws03res.dll
hnetcfg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 343.50 KB (351,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll
wshtcpip 5.2.3790.0 (srv03_rtm.030324-2048) 18.00 KB (18,432 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\wshtcpip.dll
ipsecsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 180.50 KB (184,832 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ipsecsvc.dll
oakley 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 264.00 KB (270,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\oakley.dll
winipsec 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 35.50 KB (36,352 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\winipsec.dll
pstorsvc 5.2.3790.0 (srv03_rtm.030324-2048) 24.00 KB (24,576 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\pstorsvc.dll
psbase 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 84.00 KB (86,016 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\psbase.dll
dssenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 139.98 KB (143,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\dssenh.dll
wlbsctrl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 82.00 KB (83,968 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wlbsctrl.dll
w3ssl 6.0.3790.0 (srv03_rtm.030324-2048) 15.00 KB (15,360 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\w3ssl.dll
strmfilt 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 84.00 KB (86,016 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\strmfilt.dll
httpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\httpapi.dll
svchost 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\svchost.exe
rpcss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 406.00 KB (415,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rpcss.dll
dhcpcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 113.50 KB (116,224 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\dhcpcsvc.dll
dnsrslvr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 44.50 KB (45,568 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\dnsrslvr.dll
netman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 258.50 KB (264,704 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netman.dll
netshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.73 MB (1,812,992 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\netshell.dll
clusapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 60.00 KB (61,440 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\clusapi.dll
rasapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 239.50 KB (245,248 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\rasapi32.dll
rasman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 61.50 KB (62,976 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\rasman.dll
tapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 179.50 KB (183,808 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\tapi32.dll
wininet 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 646.00 KB (661,504 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wininet.dll
wzcsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 41.00 KB (41,984 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wzcsapi.dll
wzcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 364.50 KB (373,248 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wzcsvc.dll
wmi 5.2.3790.0 (srv03_rtm.030324-2048) 6.50 KB (6,656 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\wmi.dll
ntmarta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 120.50 KB (123,392 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntmarta.dll
lmhsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.50 KB (19,968 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\lmhsvc.dll
winrnr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.00 KB (17,408 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\winrnr.dll
rasadhlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 7.50 KB (7,680 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll
winhttp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 353.00 KB (361,472 bytes) 3/24/2005 8:41 PM Microsoft Corporation c:\windows\winsxs\x86_microsoft.windows.winhttp_6595b64144ccf1df_5.1.3790.1830_x-ww_74150efb\winhttp.dll
rastls 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 180.00 KB (184,320 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rastls.dll
cryptui 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 496.50 KB (508,416 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptui.dll
raschap 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 119.50 KB (122,368 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\raschap.dll

schedsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 197.50 KB (202,240 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\schedsvc.dll
wiarpc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 32.50 KB (33,280 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wiarpc.dll
msidle 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 6.50 KB (6,656 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msidle.dll
audiosrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 40.50 KB (41,472 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\audiosrv.dll
wkssvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 130.00 KB (133,120 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\wkssvc.dll
aelupsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 26.00 KB (26,624 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\aelupsvc.dll
apphelp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 146.50 KB (150,016 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\apphelp.dll
cryptsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 55.50 KB (56,832 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptsvc.dll
certcli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 227.00 KB (232,448 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\certcli.dll
vssapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 548.00 KB (561,152 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\vssapi.dll
dmserver 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 25.50 KB (26,112 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\dmserver.dll
es 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 233.00 KB (238,592 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\es.dll
pchsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 39.00 KB (39,936 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchsvc.dll
srvsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\srvsvc.dll
sacsvr 5.2.3790.0 (srv03_rtm.030324-2048) 12.00 KB (12,288 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\sacsvr.dll
seclogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 18.50 KB (18,944 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\seclogon.dll
trkwks 5.2.3790.0 (srv03_rtm.030324-2048) 85.00 KB (87,040 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\trkwks.dll

sens 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.50 KB (37,376 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\sens.dll
wmisvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 140.00 KB (143,360 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmisvc.dll
browser 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 76.50 KB (78,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\browser.dll
wuauserv 5.7.3790.1830 (srv03_sp1_rtm.050324-1447) 8.00 KB (8,192 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\wuauserv.dll
wuaueng 5.7.3790.1830 (srv03_sp1_rtm.050324-1447) 1.18 MB (1,232,896 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\wuaueng.dll
advpack 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 98.00 KB (100,352 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\advpack.dll
cabinet 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 81.50 KB (83,456 bytes) 3/24/2005 7:35 PM Microsoft Corporation c:\windows\system32\cabinet.dll
mspatcha 5.2.3790.0 (srv03_rtm.030324-2048) 29.00 KB (29,696 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\mspatcha.dll
shfolder 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\shfolder.dll
comsvcs 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 1.19 MB (1,248,256 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\comsvcs.dll
netrap 5.2.3790.0 (srv03_rtm.030324-2048) 11.50 KB (11,776 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\netrap.dll
wbemcore 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50 KB (509,440 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 250.00 KB (256,000 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\esscli.dll
wmiutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 172.50 KB (176,640 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 404.00 KB (413,696 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 271.50 KB (278,016 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll
ncprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 46.50 KB (47,616 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll

msxml3 8.70.1104.0 1.06 MB (1,107,456 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msxml3.dll
licdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 439.00 KB (449,536 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\licdll.dll
ntlsapi 5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB (8,192 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\ntlsapi.dll
spoolsv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 57.00 KB (58,368 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\spoolsv.exe
spoolss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 85.00 KB (87,040 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\spoolss.dll
localspl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 339.00 KB (347,136 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\localspl.dll
cnbjmon 5.2.3790.1224 (dnsrv(skatari).040514-1058) 46.50 KB (47,616 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cnbjmon.dll
pjlmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 15.00 KB (15,360 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\pjlmon.dll
tcpmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\tcpmon.dll
wsnmp32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 43.00 KB (44,032 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wsnmp32.dll
tcpmib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.50 KB (17,920 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\tcpmib.dll
wsock32 5.2.3790.0 (srv03_rtm.030324-2048) 22.00 KB (22,528 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\wsock32.dll
mgmtapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 15.50 KB (15,872 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\mgmtapi.dll
snmpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.50 KB (19,968 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\snmpapi.dll
usbmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.00 KB (17,408 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\usbmon.dll
wshqos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\wshqos.dll
win32spl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 100.50 KB (102,912 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\win32spl.dll
inetpp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 75.00 KB (76,800 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\inetpp.dll

icmp 5.2.3790.0 (srv03_rtm.030324-2048) 4.50 KB (4,608 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\icmp.dll
msdtc 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 6.00 KB (6,144 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msdtc.exe
msdtctm 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 984.50 KB (1,008,128 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msdtctm.dll
msdtclog 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 73.50 KB (75,264 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msdtclog.dll
msdtcprx 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 455.50 KB (466,432 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msdtcprx.dll
mtxclu 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 77.00 KB (78,848 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mtxclu.dll
xolehlp 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 10.50 KB (10,752 bytes) 4/14/2005 10:00 AM Microsoft Corporation c:\windows\system32\xolehlp.dll
resutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 63.50 KB (65,024 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\resutils.dll
mtxoci 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 108.50 KB (111,104 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mtxoci.dll
aclient 6.1.375 4.62 MB (4,845,644 bytes) 5/3/2005 8:59 AM Altiris, Inc. c:\program files\altiris\aclient\aclient.exe

comdlg32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 274.50 KB (281,088 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\comdlg32.dll
riched32 5.2.3790.0 (srv03_rtm.030324-2048) 3.50 KB (3,584 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\riched32.dll
riched20 5.31.23.1224 439.00 KB (449,536 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\riched20.dll

ersvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ersvc.dll
regsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 68.50 KB (70,144 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\regsvc.dll
rsys Not Available 9.50 KB (9,728 bytes) 9/25/2003 6:41 PM Not Available c:\program files\benchcraft\rsys.exe
msvcr70 7.00.9466.0 336.00 KB (344,064 bytes) 1/5/2002 3:37 AM Microsoft Corporation c:\program files\benchcraft\msvcr70.dll

termsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 239.00 KB (244,736 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\termsrv.dll
icaapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 12.50 KB (12,800 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\icaapi.dll
mstlsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 116.00 KB (118,784 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mstlsapi.dll
rdpwsx 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 101.63 KB (104,072 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\rdpwsx.dll
wmiprvse 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 203.00 KB (207,872 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmiprvse.exe
faultrep 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 84.50 KB (86,528 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\faultrep.dll
wmiprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 141.00 KB (144,384 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\wmiprov.dll
logon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50 KB (509,440 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\logon.scr
dllhost 5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\dllhost.exe
txflog 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 96.50 KB (98,816 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\txflog.dll
catsrv 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 273.00 KB (279,552 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\catsrv.dll
clbcatex 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 102.50 KB (104,960 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\clbcatex.dll
inetinfo 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\inetinfo.exe
iisutil 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 164.00 KB (167,936 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\inetsrv\iisutil.dll
rpcref 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 4.00 KB (4,096 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\inetsrv\rpcref.dll
iisrtl 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 138.50 KB (141,824 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\iisrtl.dll
iisadmin 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 21.00 KB (21,504 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\inetsrv\iisadmin.dll
coadmin 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 62.50 KB (64,000 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\inetsrv\coadmin.dll

admwprox 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\admwprox.dll
iiscfg 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 1.08 MB (1,133,056 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\iiscfg.dll
metadata 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 229.00 KB (234,496 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\metadata.dll
svcext 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 43.50 KB (44,544 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\svcext.dll
security 5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\security.dll
iismap 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 58.50 KB (59,904 bytes) 4/14/2005 10:03 AM Microsoft Corporation c:\windows\system32\iismap.dll
wamreg 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 54.50 KB (55,808 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\wamreg.dll
iisw3adm 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 211.00 KB (216,064 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\iisw3adm.dll
w3cache 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 19.00 KB (19,456 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\w3cache.dll
w3tp 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 13.00 KB (13,312 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\w3tp.dll
lonsint 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 13.00 KB (13,312 bytes) 4/14/2005 10:02 AM Microsoft Corporation c:\windows\system32\inetsrv\lonsint.dll
cimwin32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.31 MB (1,372,160 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\cimwin32.dll
framedyn 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 174.50 KB (178,688 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\framedyn.dll
cfgmgr32 5.2.3790.0 (srv03_rtm.030324-2048) 17.50 KB (17,920 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\cfgmgr32.dll
licwmi 5.2.3790.0 (srv03_rtm.030324-2048) 58.50 KB (59,904 bytes) 4/13/2005 3:12 PM Microsoft Corporation c:\windows\system32\licwmi.dll
ntevt 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 230.50 KB (236,032 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\ntevt.dll
provthrd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 188.00 KB (192,512 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\wbem\provthrd.dll
msvcirt 7.0.3790.0 (srv03_rtm.030324-2048) 50.00 KB (51,200 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msvcirt.dll

msinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 376.00 KB (385,024 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\msinfo.dll
mfc42u 6.06.8063.0 1.11 MB (1,163,776 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\mfc42u.dll

odbc32 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 240.00 KB (245,760 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\odbc32.dll
odbcint 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 92.00 KB (94,208 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\odbcint.dll
cryptnet 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 61.00 KB (62,464 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cryptnet.dll
sensapi 5.2.3790.0 (srv03_rtm.030324-2048) 6.00 KB (6,144 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\sensapi.dll
drprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\drprov.dll
ntlanman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 43.50 KB (44,544 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\ntlanman.dll
netui0 5.2.3790.0 (srv03_rtm.030324-2048) 75.50 KB (77,312 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\netui0.dll
netui1 5.2.3790.0 (srv03_rtm.030324-2048) 184.00 KB (188,416 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\netui1.dll
davclnt 5.2.3790.0 (srv03_rtm.030324-2048) 23.50 KB (24,064 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\davclnt.dll

[Services]

Display Name Name State Start Mode Service Type Path Error Control Start Name Tag ID
Altiris Client Service AClient Running Auto Own Process c:\program files\altiris\aclient\aclient.exe -service Normal LocalSystem 0
Application Experience Lookup Service AeLookupSvc Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Alerter Alerter Stopped Disabled Share Process c:\windows\system32\svchost.exe -k localservice Normal NT AUTHORITY\LocalService 0
Application Layer Gateway Service ALG Stopped Manual Own Process c:\windows\system32\alg.exe Normal NT AUTHORITY\LocalService 0
Application Management AppMgmt Stopped Manual Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Windows Audio AudioSrv Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Background Intelligent Transfer Service BITS Stopped Manual Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Computer Browser Browser Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Indexing Service CiSvc Stopped Disabled Share Process c:\windows\system32\cisvc.exe Normal LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process c:\windows\system32\clipsrv.exe Normal LocalSystem 0
COM+ System Application COMSysApp Running Manual Own Process c:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235} Normal LocalSystem 0
Cryptographic Services CryptSvc Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
DCOM Server Process Launcher DcomLaunch Running Auto Share Process c:\windows\system32\svchost.exe -k dcomlaunch Normal LocalSystem 0
Distributed File System Dfs Stopped Manual Own Process c:\windows\system32\dfssvc.exe Normal LocalSystem 0
DHCP Client Dhcp Running Auto Share Process c:\windows\system32\svchost.exe -k networkservice Normal NT AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service dmadmin Stopped Manual Share Process c:\windows\system32\dmadmin.exe /com Normal LocalSystem 0
Logical Disk Manager dmserver Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
DNS Client Dnscache Running Auto Share Process c:\windows\system32\svchost.exe -k networkservice Normal NT AUTHORITY\NetworkService 0
Error Reporting Service ERSvc Running Auto Share Process c:\windows\system32\svchost.exe -k winerr Ignore LocalSystem 0
Event Log Eventlog Running Auto Share Process c:\windows\system32\services.exe Normal LocalSystem 0
COM+ Event System EventSystem Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Help and Support helpsvc Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal
Human Interface Device Access HidServ Stopped Disabled Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
HTTP SSL HTTPFilter Running Manual Share Process c:\windows\system32\lsass.exe Normal LocalSystem 0
IIS Admin Service IISADMIN Running Auto Share Process c:\windows\system32\inetsrv\inetinfo.exe Normal LocalSystem 0
IMAPI CD-Burning COM Service ImapiService Stopped Disabled Own Process c:\windows\system32\imapi.exe Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process c:\windows\system32\ismserv.exe Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled Share Process c:\windows\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
License Logging LicenseService Stopped Disabled Own Process c:\windows\system32\llssrv.exe Normal NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Running Auto Share Process c:\windows\system32\svchost.exe -k localservice Normal NT AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Disabled Own Process c:\windows\system32\mnmsrvc.exe Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Own Process c:\windows\system32\msdtc.exe Normal NT AUTHORITY\NetworkService 1
Windows Installer MSIServer Stopped Manual Share Process c:\windows\system32\msiexec.exe /v Normal LocalSystem 0
Network DDE NetDDE Stopped Disabled Share Process c:\windows\system32\netdde.exe Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped Disabled Share Process c:\windows\system32\netdde.exe Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process c:\windows\system32\lsass.exe Normal LocalSystem 0

```
Network Connections Netman Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Network Location Awareness (NLA) Nla Stopped Manual
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
File Replication NtFrs Stopped Manual Own
Process c:\windows\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped Manual
Share Process c:\windows\system32\lsass.exe
Normal LocalSystem 0
Removable Storage NtmsSvc Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Plug and Play PlugPlay Running Auto Share Process
c:\windows\system32\services.exe Normal
LocalSystem 0
IPSEC Services PolicyAgent Running Auto Share
Process c:\windows\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Running Auto
Share Process c:\windows\system32\lsass.exe
Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Access Connection Manager RasMan Stopped Manual
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr Stopped Manual
Own Process c:\windows\system32\sessmgr.exe Normal
LocalSystem 0
Routing and Remote Access RemoteAccess Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Registry RemoteRegistry Running Auto
Share Process c:\windows\system32\svchost.exe -k
regsvc Normal NT AUTHORITY\LocalService 0

Remote Command Service RMSYS Running Auto Own
Process "c:\program files\benchcraft\rsys.exe" Normal
.\Administrator 0
Remote Procedure Call (RPC) Locator RpcLocator Stopped Manual
Own Process c:\windows\system32\locator.exe Normal
NT AUTHORITY\NetworkService 0
Remote Procedure Call (RPC) RpcSs Running Auto
Share Process c:\windows\system32\svchost.exe -k
rpcss Normal NT Authority\NetworkService 0

Resultant Set of Policy Provider RSoPProv Stopped Manual
Share Process c:\windows\system32\rsopprov.exe
Special Administration Console Helper sacsvr Running Manual
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share
Process c:\windows\system32\lsass.exe Normal
LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
c:\windows\system32\scardsvr.exe Ignore NT
AUTHORITY\LocalService 0
Task Scheduler Schedule Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Secondary Logon seclogon Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Ignore
LocalSystem 0
System Event Notification SENS Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Windows Firewall/Internet Connection Sharing (ICS) SharedAccess
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Shell Hardware Detection ShellHWDetection Running Auto
Share Process c:\windows\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Print Spooler Spooler Running Auto Own Process
c:\windows\system32\spoolsv.exe Normal
LocalSystem 0
Windows Image Acquisition (WIA) stisvc Stopped Disabled
Share Process c:\windows\system32\svchost.exe -k
imgsvc Normal NT AUTHORITY\LocalService 0

Microsoft Software Shadow Copy Provider swprv Stopped
Manual Own Process c:\windows\system32\svchost.exe -k
swprv Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Manual
Own Process c:\windows\system32\smlogsvc.exe Normal
NT Authority\NetworkService 0
Telephony TapiSrv Stopped Manual Share Process
c:\windows\system32\svchost.exe -k tapisrv Normal
LocalSystem 0
Terminal Services TermService Running Manual Share
Process c:\windows\system32\svchost.exe -k termsvcs Normal
LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows\system32\tlntsvr.exe Normal NT
AUTHORITY\LocalService 0

Distributed Link Tracking Server TrkSvr Stopped Disabled
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Terminal Services Session Directory Tssdis Stopped Disabled
Own Process c:\windows\system32\tssdis.exe Normal
LocalSystem 0
Windows User Mode Driver Framework UMWdf Stopped Manual
Own Process c:\windows\system32\wdfmgr.exe Normal
NT AUTHORITY\LocalService 0
Upload Manager uploadmgr Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Uninterruptible Power Supply UPS Stopped Manual
Own Process c:\windows\system32\ups.exe Normal
NT AUTHORITY\LocalService 0
Virtual Disk Service vds Stopped Manual Own
Process c:\windows\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\vssvc.exe Normal
LocalSystem 0
Windows Time W32Time Running Auto Share
Process c:\windows\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
World Wide Web Publishing Service W3SVC Running Auto
Share Process c:\windows\system32\svchost.exe -k
iissvcs Normal LocalSystem 0
WebClient WebClient Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service WinHttpAutoProxySvc
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k localservice Normal
NT AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt Running Auto
Share Process c:\windows\system32\svchost.exe -k
netsvcs Ignore LocalSystem 0
Portable Media Serial Number Service WmdmPmSN Stopped Manual
Share Process c:\windows\system32\svchost.exe -k
netsvcs Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped Manual Own
Process c:\windows\system32\wbem\wmiapsrv.exe Normal
LocalSystem 0
Automatic Updates wuauserv Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
```

Wireless Configuration WZCSVC Running Auto Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Network Provisioning Service xmlprov Stopped Manual Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0

[Program Groups]

Group Name Name User Name
Accessories Default User:Accessories Default User
Accessories\Accessibility Default User:Accessories\Accessibility Default User
Accessories\Entertainment Default User:Accessories\Entertainment Default User
Startup Default User:Startup Default User
Accessories All Users:Accessories All Users
Accessories\Accessibility All Users:Accessories\Accessibility All Users

Accessories\Communications All Users:Accessories\Communications All Users
Accessories\Entertainment All Users:Accessories\Entertainment All Users

Accessories\System Tools All Users:Accessories\System Tools All Users

Administrative Tools All Users:Administrative Tools All Users

Microsoft Network Monitor All Users:Microsoft Network Monitor All Users

Microsoft SQL Server All Users:Microsoft SQL Server All Users

Microsoft Visual Studio .NET 2003 All Users:Microsoft Visual Studio .NET 2003 All Users
Microsoft Visual Studio .NET 2003\Visual Studio .NET Tools All Users:Microsoft Visual Studio .NET 2003\Visual Studio .NET Tools All Users

Startup All Users:Startup All Users
Sysinternals PsTools All Users:Sysinternals PsTools All Users

Accessories NT AUTHORITY\SYSTEM:Accessories NT AUTHORITY\SYSTEM
Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility NT AUTHORITY\SYSTEM
Accessories\Entertainment NT AUTHORITY\SYSTEM:Accessories\Entertainment NT AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT AUTHORITY\SYSTEM
Accessories DL1\Administrator:Accessories DL1\Administrator

Accessories\Accessibility DL1\Administrator:Accessories\Accessibility DL1\Administrator

Accessories\Communications DL1\Administrator:Accessories\Communications DL1\Administrator
Accessories\Communications\HyperTerminal DL1\Administrator:Accessories\Communications\HyperTerminal DL1\Administrator
Accessories\Entertainment DL1\Administrator:Accessories\Entertainment DL1\Administrator
Administrative Tools DL1\Administrator:Administrative Tools DL1\Administrator
Benchcraft DL1\Administrator:Benchcraft DL1\Administrator

QLogic Corporation DL1\Administrator:QLogic Corporation DL1\Administrator
QLogic Corporation\SANblade Control VIX DL1\Administrator:QLogic Corporation\SANblade Control VIX DL1\Administrator
Startup DL1\Administrator:Startup DL1\Administrator

[Startup Programs]

Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM Startup
desktop desktop.ini DL1\Administrator Startup
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common Startup
AClntUsr c:\program files\altiris\aclient\aclntusr.exe All Users HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
WordPad Document "%programfiles%\windows nt\accessories\wordpad.exe"
Windows Media Services DRM Storage object Not Available
Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details

[Internet Settings]

[Internet Explorer]

[Summary]
[Following are sub-categories of this main category]

Item Value
Version 6.0.3790.1830
Build 63790.1830
Application Path C:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File Version Size Date Path Company

actxprxy.dll 6.0.3790.1830 97 KB 3/24/2005 4:55:26 PM C:\WINDOWS\system32 Microsoft Corporation
advpack.dll 6.0.3790.1830 98 KB 3/24/2005 4:55:28 PM C:\WINDOWS\system32 Microsoft Corporation
asctrls.ocx 6.0.3790.0 90 KB 3/25/2003 4:00:00 AM C:\WINDOWS\system32 Microsoft Corporation
browselc.dll 6.0.3790.0 62 KB 3/25/2003 4:00:00 AM C:\WINDOWS\system32 Microsoft Corporation
browseui.dll 6.0.3790.1830 1,009 KB 3/24/2005 4:56:10 PM C:\WINDOWS\system32 Microsoft Corporation
cdfview.dll 6.0.3790.1830 149 KB 3/24/2005 4:56:32 PM C:\WINDOWS\system32 Microsoft Corporation
comctl32.dll 5.82.3790.1830 585 KB 3/24/2005 4:57:56 PM C:\WINDOWS\system32 Microsoft Corporation
dxtrans.dll 6.3.3790.1830 205 KB 3/24/2005 5:00:58 PM C:\WINDOWS\system32 Microsoft Corporation
dxtmsft.dll 6.3.3790.1830 355 KB 3/24/2005 5:00:58 PM C:\WINDOWS\system32 Microsoft Corporation
iecont.dll <File Missing> Not Available Not Available Not Available Not Available
iecontlc.dll <File Missing> Not Available Not Available Not Available Not Available
iedkcs32.dll 16.0.3790.1830 324 KB 3/24/2005 5:04:58 PM C:\WINDOWS\system32 Microsoft Corporation
iepeers.dll 6.0.3790.1830 248 KB 3/24/2005 5:04:58 PM C:\WINDOWS\system32 Microsoft Corporation
iesetup.dll 6.0.3790.1830 61 KB 3/24/2005 5:04:58 PM C:\WINDOWS\system32 Microsoft Corporation
ieuinit.inf Not Available 24 KB 3/24/2005 5:04:58 PM C:\WINDOWS\system32 Not Available
iexplore.exe 6.0.3790.1830 92 KB 3/24/2005 5:04:58 PM C:\Program Files\Internet Explorer Microsoft Corporation

imgutil.dll 6.0.3790.1830 38 KB 3/24/2005 5:05:04 PM C:\WINDOWS\system32 Microsoft Corporation

inetcpl.cpl 6.0.3790.1830 358 KB 3/24/2005 5:05:06 PM
C:\WINDOWS\system32 Microsoft Corporation
inetcplc.dll 6.0.3790.0 109 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
inseng.dll 6.0.3790.1830 94 KB 3/24/2005 5:05:06 PM
C:\WINDOWS\system32 Microsoft Corporation
mlang.dll 6.0.3790.1830 578 KB 3/24/2005 5:07:20 PM
C:\WINDOWS\system32 Microsoft Corporation
msencode.dll 2002.10.4.0 112 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 □□□▪ □v□□
mshta.exe 6.0.3790.1830 30 KB 3/24/2005 5:07:26 PM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll 6.0.3790.1830 3,036 KB 3/24/2005 5:07:26 PM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.tlb 6.0.3790.1830 1,320 KB 3/24/2005 5:07:26 PM
C:\WINDOWS\system32 Microsoft Corporation
mshtmled.dll 6.0.3790.1830 455 KB 3/24/2005 5:07:26 PM
C:\WINDOWS\system32 Microsoft Corporation
mshtmler.dll 6.0.3790.1830 56 KB 3/24/2005 5:07:26 PM
C:\WINDOWS\system32 Microsoft Corporation
msident.dll 6.0.3790.1830 48 KB 3/24/2005 5:07:28 PM
C:\WINDOWS\system32 Microsoft Corporation
msidntld.dll 6.0.3790.0 15 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
msieftp.dll 6.0.3790.1830 244 KB 3/24/2005 5:07:28 PM
C:\WINDOWS\system32 Microsoft Corporation
msrating.dll 6.0.3790.1830 144 KB 3/24/2005 5:07:36 PM
C:\WINDOWS\system32 Microsoft Corporation
mstime.dll 6.0.3790.1830 523 KB 3/24/2005 5:07:38 PM
C:\WINDOWS\system32 Microsoft Corporation
occache.dll 6.0.3790.1830 94 KB 3/24/2005 5:08:34 PM
C:\WINDOWS\system32 Microsoft Corporation
proctexe.ocx 6.3.3790.1830 83 KB 3/24/2005 5:12:26 PM
C:\WINDOWS\system32 Intel Corporation
sendmail.dll 6.0.3790.1830 56 KB 3/24/2005 5:13:36 PM
C:\WINDOWS\system32 Microsoft Corporation
shdoclc.dll 6.0.3790.0 589 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll 6.0.3790.1830 1,468 KB 3/24/2005 5:13:36 PM
C:\WINDOWS\system32 Microsoft Corporation
shfolder.dll 6.0.3790.1830 25 KB 3/24/2005 5:13:36 PM
C:\WINDOWS\system32 Microsoft Corporation
shlwapi.dll 6.0.3790.1830 314 KB 3/24/2005 5:13:40 PM
C:\WINDOWS\system32 Microsoft Corporation
tdc.ocx 1.3.0.3130 58 KB 3/25/2003 4:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
url.dll 6.0.3790.1830 37 KB 3/24/2005 5:26:12 PM
C:\WINDOWS\system32 Microsoft Corporation
urlmon.dll 6.0.3790.1830 673 KB 3/24/2005 5:26:12 PM
C:\WINDOWS\system32 Microsoft Corporation
webcheck.dll 6.0.3790.1830 273 KB 3/24/2005 5:26:16 PM
C:\WINDOWS\system32 Microsoft Corporation
wininet.dll 6.0.3790.1830 646 KB 3/24/2005 5:26:18 PM
C:\WINDOWS\system32 Microsoft Corporation

[Connectivity]

Item Value
Connection Preference Never dial

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Disabled
AutoConfigURL
Proxy Disabled
ProxyServer itgproxy:80
ProxyOverride

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder C:\Documents and Settings\Administrator.SQLTPCKIT\Local Settings\Temporary Internet Files

Total Disk Space Not Available
Available Disk Space Not Available
Maximum Cache Size Not Available
Available Cache Size Not Available

[List of Objects]

Program File Status CodeBase
{9F1C11AA-197B-4942-BA54-47A8489BB47F} Not Available
http://v4.windowsupdate.microsoft.com/CAB/x86/unicode/iuctl.CAB?38455.6793981481

[Content]

[Following are sub-categories of this main category]
[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranet Custom
Trusted sites Custom
Internet Custom
Restricted sites Custom

COM+ Settings

TPCC.AllTxns:
Activation:
Enable Object Pooling selected
Minimum Pool Size: 55
Maximum Pool Size: 55
Creation Timeout: 60000
Enable Object Construction
Enable Just In Time Activation
Concurrency:
Concurrency Required

Microsoft IIS Registry Parameters

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00008ca0
"PoolThreadLimit"=dword:00001ffc

```
"MaxPoolThreads"=dword:00000ffe
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"PerfIniFile"="infoctrs.ini"
"Last Counter"=dword:00001202
"Last Help"=dword:00001203
"First Counter"=dword:000011c2
"First Help"=dword:000011c3
"Object List"="4546"
"Library Validation
Code"=hex:00,b5,b9,10,50,4b,c5,01,00,20,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:4c,c3,d3,e7,44,ca,56,e0,f3,e8,a0,14,52,26,fb,
0f
"WbemAdapFileTime"=hex:0c,a5,68,10,50,4b,c5,01
"WbemAdapFileSize"=dword:00002000
"WbemAdapStatus"=dword:00000000
```

World Wide Web Service Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00
,6f,00,\

74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73
,\

00,76,00,63,00,68,00,6f,00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00
,\
  6b,00,20,00,69,00,69,00,73,00,73,00,76,00,63,00,73,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):52,00,50,00,43,00,53,00,53,00,00,00,48,00,54,0
0,54,00,\

50,00,46,00,69,00,6c,00,74,00,65,00,72,00,00,00,49,00,49,00,53,00,41,00,4
4,\
  00,4d,00,49,00,4e,00,00,00,00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the
Internet Information Services Manager"
"FailureActions"=hex:80,51,01,00,00,00,00,00,00,00,00,03,00,00,00,14,00
,00,\

00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,0
0

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000006
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINDOWS\\system32\\inetsrv"
"AccessDeniedMessage"="Error: Access is Denied."
"ServiceDll"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,44,00,4f,00,57,00,5
3,\

00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00
,\

65,00,74,00,73,00,72,00,76,00,5c,00,69,00,69,00,73,00,77,00,33,00,61,00,6
4,\
  00,6d,00,2e,00,64,00,6c,00,6c,00,00,00
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Library"="C:\\WINDOWS\\system32\\inetsrv\\w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"PerfIniFile"="w3ctrs.ini"
"Last Counter"=dword:000012fa
"Last Help"=dword:000012fb
"First Counter"=dword:00001204
"First Help"=dword:00001205
"Object List"="4612 4786"
"Library Validation
Code"=hex:00,b5,b9,10,50,4b,c5,01,00,5e,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:39,e3,6c,2c,b4,be,59,f5,17,7c,c4,d5,2f,dc,f7,
1a
"WbemAdapFileTime"=hex:c8,18,9d,10,50,4b,c5,01
"WbemAdapFileSize"=dword:00005e00
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,0
0,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,
\
00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00
,\
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,
\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,04,00,00,00,0
0,\

00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,06,00,00,00,00,00,14,00,00,0
1,\

00,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,00,18,00,fd,01,02,00,01,02,00
,\

00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,12,00,00,0
0,\
  01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

IIS6 Configuration Script

```
@ECHO OFF
@cd %SystemRoot%\System32
@cscript %SystemRoot%\System32\iisext.vbs /AddFile
C:\INETPUB\WWWROOT\tpcc.dll 1 TPCC 0 TPCC DLL
                >   IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i /y
/q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/LogType 0
                >>  IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i /y
/q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessExecute True                                 >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessRead True                                    >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
```

```
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessScript True                                  >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AuthAnonymous True                                 >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AuthNTLM True
        >> IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/DefaultAppPool/AppPoolIdentityType 1             >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartTime 0                            >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartRequests 0                        >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartMemory 0                          >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleTime False                         >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleRequests False                     >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleSchedule False                     >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
```

```
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleMemory False                       >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleIsapiUnhealthy False               >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleOnDemand False     >> IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleConfigChange False >> IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecyclePrivateMemory False>> IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PingingEnabled False                             >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/RapidFailProtection False                        >>
IIS6_CONFIG.out
@If %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.ERR+IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
```

TPCC Application Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="C:\\Inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:00000008
"MaxConnections"=dword:00008ca0
"MaxPendingDeliveries"=dword:000009c4
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="via:sqlorca,1441,1"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="ssdl"
```

```
"ConnectDelay"=dword:0000000a
COM_SingleProc=yes
"CallNoDuplicatesNewOrder"=dword:00000000
"spprefix"="sp0"
```

NOTE: This is representative of 1 web client. Spprefix was varied from 0-15 by every 2 web clients. DBServer was varied on every web clients to connect to the appropriate SoftNuma node

TCPIP Registry Parameters

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Paramet
ers]
"NV Hostname"="dl1"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,
00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00
,\
  64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="dl1"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"NV Domain"=""
"MaxUserPort"=dword:0000fffe
"DhcpNameServer"="15.1.101.1"
"DhcpDomain"="hp-perf.net"
```

RTE Input Parameters

3Tier.pro

Profile: Perf-3-Tier-99350
File Path: D:\1.2m_audit\PerfRun11\Perf-3-Tier-99350.xml
Version: 5

Number of Engines: 92

Name: DRIVER.01
Description:
Directory: C:\DRIVER.1.log
Machine: CL1
Parameter Set: Audit
Index: 0
Seed: 73660
Configured Users: 13000
Pipe Name: DRIVER.1
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp1

Name: DRIVER.02
Description:
Directory: C:\DRIVER.2.log
Machine: CL1
Parameter Set: Audit
Index: 22500000
Seed: 73660
Configured Users: 13000
Pipe Name: DRIVER.2
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp1

Name: DRIVER.03
Description:
Directory: C:\DRIVER.3.log
Machine: CL2
Parameter Set: Audit
Index: 45000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.3
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp2

Name: DRIVER.04
Description:
Directory: C:\DRIVER.4.log
Machine: CL2
Parameter Set: Audit
Index: 67500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.4
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp2

Name: DRIVER.05
Description:
Directory: C:\DRIVER.5.log
Machine: CL33
Parameter Set: Audit
Index: 90000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.5
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp2

Name: DRIVER.06
Description:
Directory: C:\DRIVER.6.log
Machine: CL3
Parameter Set: Audit
Index: 112500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.6
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp3

Name: DRIVER.07
Description:
Directory: C:\DRIVER.7.log
Machine: CL3
Parameter Set: Audit
Index: 135000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.7
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp3

Name: DRIVER.08
Description:
Directory: C:\DRIVER.8.log
Machine: CL33
Parameter Set: Audit
Index: 157500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.8
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp3

Name: DRIVER.09
Description:
Directory: C:\DRIVER.9.log
Machine: CL4
Parameter Set: Audit
Index: 180000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.9
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp4

Name: DRIVER.10
Description:
Directory: C:\DRIVER.10.log
Machine: CL4
Parameter Set: Audit
Index: 202500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.10
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp4

Name: DRIVER.11
Description:
Directory: C:\DRIVER.11.log
Machine: CL33
Parameter Set: Audit
Index: 225000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.11
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp4

Name: DRIVER.12
Description:
Directory: C:\DRIVER.12.log
Machine: CL5
Parameter Set: Audit
Index: 247500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.12
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp5

Name: DRIVER.13
Description:
Directory: C:\DRIVER.13.log
Machine: CL5
Parameter Set: Audit
Index: 270000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.13
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp5

Name: DRIVER.14
Description:
Directory: C:\DRIVER.14.log
Machine: CL33
Parameter Set: Audit
Index: 292500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.14
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp5

Name: DRIVER.15
Description:
Directory: C:\DRIVER.15.log
Machine: CL6
Parameter Set: Audit
Index: 315000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.15
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp6

Name: DRIVER.16
Description:
Directory: C:\DRIVER.16.log
Machine: CL6
Parameter Set: Audit
Index: 337500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.16
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp6

Name: DRIVER.17
Description:
Directory: C:\DRIVER.17.log
Machine: CL33
Parameter Set: Audit
Index: 360000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.17
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp6

Name: DRIVER.18
Description:
Directory: C:\DRIVER.18.log
Machine: CL7
Parameter Set: Audit
Index: 382500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.18
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp7

Name: DRIVER.19
Description:
Directory: C:\DRIVER.19.log
Machine: CL7
Parameter Set: Audit
Index: 405000000

Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.19
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp7

Name: DRIVER.20
Description:
Directory: C:\DRIVER.20.log
Machine: CL33
Parameter Set: Audit
Index: 427500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.20
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp7

Name: DRIVER.21
Description:
Directory: C:\DRIVER.21.log
Machine: CL8
Parameter Set: Audit
Index: 450000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.21
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp8

Name: DRIVER.22
Description:
Directory: C:\DRIVER.22.log
Machine: CL8
Parameter Set: Audit
Index: 472500000
Seed: 73660

Pipe Name: DRIVER.22
Configured Users: 15000
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp8

Name: DRIVER.23
Description:
Directory: C:\DRIVER.23.log
Machine: CL33
Parameter Set: Audit
Index: 495000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.23
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp8

Name: DRIVER.24
Description:
Directory: C:\DRIVER.24.log
Machine: CL9
Parameter Set: Audit
Index: 517500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.24
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp9

Name: DRIVER.25
Description:
Directory: C:\DRIVER.25.log
Machine: CL9
Parameter Set: Audit
Index: 540000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.25
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp9

Name: DRIVER.26
Description:
Directory: C:\DRIVER.26.log
Machine: CL34
Parameter Set: Audit
Index: 562500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.26
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp9

Name: DRIVER.27
Description:
Directory: C:\DRIVER.27.log
Machine: CL10
Parameter Set: Audit
Index: 585000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.27
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp10

Name: DRIVER.28
Description:
Directory: C:\DRIVER.28.log
Machine: CL10
Parameter Set: Audit
Index: 607500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.28
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10

CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp10

Name: DRIVER.29
Description:
Directory: C:\DRIVER.29.log
Machine: CL34
Parameter Set: Audit
Index: 630000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.29
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp10

Name: DRIVER.30
Description:
Directory: C:\DRIVER.30.log
Machine: CL11
Parameter Set: Audit
Index: 652500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.30
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp11

Name: DRIVER.31
Description:
Directory: C:\DRIVER.31.log
Machine: CL11
Parameter Set: Audit
Index: 675000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.31
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp11

Name: DRIVER.32
Description:
Directory: C:\DRIVER.32.log
Machine: CL34
Parameter Set: Audit
Index: 697500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.32
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp11

Name: DRIVER.33
Description:
Directory: C:\DRIVER.33.log
Machine: CL12
Parameter Set: Audit
Index: 720000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.33
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp12

Name: DRIVER.34
Description:
Directory: C:\DRIVER.34.log
Machine: CL12
Parameter Set: Audit
Index: 742500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.34
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp12

Name: DRIVER.35
Description:
Directory: C:\DRIVER.35.log
Machine: CL34
Parameter Set: Audit
Index: 765000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.35
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp12

Name: DRIVER.36
Description:
Directory: C:\DRIVER.36.log
Machine: CL13
Parameter Set: Audit
Index: 787500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.36
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp13

Name: DRIVER.37
Description:
Directory: C:\DRIVER.37.log
Machine: CL13
Parameter Set: Audit
Index: 810000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.37
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp13

Name: DRIVER.38
Description:
Directory: C:\DRIVER.38.log

Machine: CL34
Parameter Set: Audit
Index: 832500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.38
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp13

Name: DRIVER.39
Description:
Directory: C:\DRIVER.39.log
Machine: CL14
Parameter Set: Audit
Index: 855000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.39
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp14

Name: DRIVER.40
Description:
Directory: C:\DRIVER.40.log
Machine: CL14
Parameter Set: Audit
Index: 877500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.40
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp14

Name: DRIVER.41
Description:
Directory: C:\DRIVER.41.log
Machine: CL34
Parameter Set: Audit
Index: 900000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.41
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp14

Name: DRIVER.42
Description:
Directory: C:\DRIVER.42.log
Machine: CL15
Parameter Set: Audit
Index: 922500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.42
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp15

Name: DRIVER.43
Description:
Directory: C:\DRIVER.43.log
Machine: CL15
Parameter Set: Audit
Index: 945000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.43
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp15

Name: DRIVER.44
Description:
Directory: C:\DRIVER.44.log
Machine: CL34
Parameter Set: Audit
Index: 967500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.44
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp15

Name: DRIVER.45
Description:
Directory: C:\DRIVER.45.log
Machine: CL16
Parameter Set: Audit
Index: 990000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.45
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp16

Name: DRIVER.46
Description:
Directory: C:\DRIVER.46.log
Machine: CL16
Parameter Set: Audit
Index: 1012500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.46
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp16

Name: DRIVER.47
Description:
Directory: C:\DRIVER.47.log
Machine: CL34
Parameter Set: Audit
Index: 1035000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.47
Connect Rate: 20

Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp16

Name: DRIVER.48
Description:
Directory: C:\DRIVER.48.log
Machine: CL17
Parameter Set: Audit
Index: 1057500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.48
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp17

Name: DRIVER.49
Description:
Directory: C:\DRIVER.49.log
Machine: CL17
Parameter Set: Audit
Index: 1080000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.49
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp17

Name: DRIVER.50
Description:
Directory: C:\DRIVER.50.log
Machine: CL35
Parameter Set: Audit
Index: 1102500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.50
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp17

Name: DRIVER.51
Description:
Directory: C:\DRIVER.51.log
Machine: CL18
Parameter Set: Audit
Index: 1125000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.51
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp18

Name: DRIVER.52
Description:
Directory: C:\DRIVER.52.log
Machine: CL18
Parameter Set: Audit
Index: 1147500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.52
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp18

Name: DRIVER.53
Description:
Directory: C:\DRIVER.53.log
Machine: CL35
Parameter Set: Audit
Index: 1170000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.53
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp18

Name: DRIVER.54
Description:
Directory: C:\DRIVER.54.log
Machine: CL19
Parameter Set: Audit
Index: 1192500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.54
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp19

Name: DRIVER.55
Description:
Directory: C:\DRIVER.55.log
Machine: CL19
Parameter Set: Audit
Index: 1215000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.55
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp19

Name: DRIVER.56
Description:
Directory: C:\DRIVER.56.log
Machine: CL35
Parameter Set: Audit
Index: 1237500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.56
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp19

Name: DRIVER.57
Description:
Directory: C:\DRIVER.57.log
Machine: CL20
Parameter Set: Audit
Index: 1260000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.57
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp20

Name: DRIVER.58
Description:
Directory: C:\DRIVER.58.log
Machine: CL20
Parameter Set: Audit
Index: 1282500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.58
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp20

Name: DRIVER.59
Description:
Directory: C:\DRIVER.59.log
Machine: CL35
Parameter Set: Audit
Index: 1305000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.59
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp20

Name: DRIVER.60
Description:
Directory: C:\DRIVER.60.log
Machine: CL21
Parameter Set: Audit
Index: 1327500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.60
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp21

Name: DRIVER.61
Description:
Directory: C:\DRIVER.61.log
Machine: CL21
Parameter Set: Audit
Index: 1350000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.61
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp21

Name: DRIVER.62
Description:
Directory: C:\DRIVER.62.log
Machine: CL35
Parameter Set: Audit
Index: 1372500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.62
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp21

Name: DRIVER.63
Description:
Directory: C:\DRIVER.63.log
Machine: CL22
Parameter Set: Audit
Index: 1395000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.63
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp22

Name: DRIVER.64
Description:
Directory: C:\DRIVER.64.log
Machine: CL22
Parameter Set: Audit
Index: 1417500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.64
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp22

Name: DRIVER.65
Description:
Directory: C:\DRIVER.65.log
Machine: CL35
Parameter Set: Audit
Index: 1440000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.65
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp22

Name: DRIVER.66
Description:
Directory: C:\DRIVER.66.log
Machine: CL23
Parameter Set: Audit
Index: 1462500000
Seed: 73660

Configured Users: 15000
Pipe Name: DRIVER.66
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp23

Name: DRIVER.67
Description:
Directory: C:\DRIVER.67.log
Machine: CL23
Parameter Set: Audit
Index: 1485000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.67
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp23

Name: DRIVER.68
Description:
Directory: C:\DRIVER.68.log
Machine: CL35
Parameter Set: Audit
Index: 1507500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.68
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp23

Name: DRIVER.69
Description:
Directory: C:\DRIVER.69.log
Machine: CL24
Parameter Set: Audit
Index: 1530000000
Seed: 73660
Configured Users: 15000
Connect Rate: 20
Pipe Name: DRIVER.69
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp24

Name: DRIVER.70
Description:
Directory: C:\DRIVER.70.log
Machine: CL24
Parameter Set: Audit
Index: 1552500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.70
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp24

Name: DRIVER.71
Description:
Directory: C:\DRIVER.71.log
Machine: CL35
Parameter Set: Audit
Index: 1575000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.71
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp24

Name: DRIVER.72
Description:
Directory: C:\DRIVER.72.log
Machine: CL25
Parameter Set: Audit
Index: 1597500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.72
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp25

Name: DRIVER.73
Description:
Directory: C:\DRIVER.73.log
Machine: CL25
Parameter Set: Audit
Index: 1620000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.73
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp25

Name: DRIVER.74
Description:
Directory: C:\DRIVER.74.log
Machine: CL36
Parameter Set: Audit
Index: 1642500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.74
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp25

Name: DRIVER.75
Description:
Directory: C:\DRIVER.75.log
Machine: CL26
Parameter Set: Audit
Index: 1665000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.75
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233

CPU: 0
Additional Options: spprefix=sp26

Name: DRIVER.76
Description:
Directory: C:\DRIVER.76.log
Machine: CL26
Parameter Set: Audit
Index: 1687500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.76
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp26

Name: DRIVER.77
Description:
Directory: C:\DRIVER.77.log
Machine: CL36
Parameter Set: Audit
Index: 1710000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.77
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp26

Name: DRIVER.78
Description:
Directory: C:\DRIVER.78.log
Machine: CL27
Parameter Set: Audit
Index: 1732500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.78
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp27

Name: DRIVER.79
Description:
Directory: C:\DRIVER.79.log
Machine: CL27
Parameter Set: Audit
Index: 1755000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.79
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp27

Name: DRIVER.80
Description:
Directory: C:\DRIVER.80.log
Machine: CL36
Parameter Set: Audit
Index: 1777500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.80
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp27

Name: DRIVER.81
Description:
Directory: C:\DRIVER.81.log
Machine: CL28
Parameter Set: Audit
Index: 1800000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.81
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp28

Name: DRIVER.82
Description:
Directory: C:\DRIVER.82.log
Machine: CL28
Parameter Set: Audit
Index: 1822500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.82
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp28

Name: DRIVER.83
Description:
Directory: C:\DRIVER.83.log
Machine: CL36
Parameter Set: Audit
Index: 1845000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.83
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp28

Name: DRIVER.84
Description:
Directory: C:\DRIVER.84.log
Machine: CL29
Parameter Set: Audit
Index: 1867500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.84
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp29

Name: DRIVER.85
Description:
Directory: C:\DRIVER.85.log
Machine: CL29

Parameter Set: Audit
Index: 1890000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.85
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp29

Name: DRIVER.86
Description:
Directory: C:\DRIVER.86.log
Machine: CL36
Parameter Set: Audit
Index: 1912500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.86
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp29

Name: DRIVER.87
Description:
Directory: C:\DRIVER.87.log
Machine: CL30
Parameter Set: Audit
Index: 1935000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.87
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp30

Name: DRIVER.88
Description:
Directory: C:\DRIVER.88.log
Machine: CL30
Parameter Set: Audit
Index: 1957500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.88
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp30

Name: DRIVER.89
Description:
Directory: C:\DRIVER.89.log
Machine: CL36
Parameter Set: Audit
Index: 1980000000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.89
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp30

Name: DRIVER.90
Description:
Directory: C:\DRIVER.90.log
Machine: CL31
Parameter Set: Audit
Index: 2002500000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.90
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp31

Name: DRIVER.91
Description:
Directory: C:\DRIVER.91.log
Machine: CL31
Parameter Set: Audit
Index: 2025000000
Seed: 73660
Configured Users: 15000
Pipe Name: DRIVER.91
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp31

Name: DRIVER.92
Description:
Directory: C:\DRIVER.92.log
Machine: CL36
Parameter Set: Audit
Index: 2047500000
Seed: 73660
Configured Users: 2250
Pipe Name: DRIVER.92
Connect Rate: 20
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp31

Number of User groups: 92

Driver Engine: DRIVER.01
IIS Server: DL1
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 1300
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 13000
District id: 1
Scale Down: No

Driver Engine: DRIVER.02
IIS Server: DL1
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1301 - 2600
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 13000
District id: 1

Scale Down: No

Driver Engine: DRIVER.03
IIS Server: DL2
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2601 - 4100
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.04
IIS Server: DL2
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4101 - 5600
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.05
IIS Server: DL2
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5601 - 5825
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.06
IIS Server: DL3
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5826 - 7325
w_id Min Warehouse: 1
Scale: Normal
w_id Max Warehouse: 99350
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.07
IIS Server: DL3
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7326 - 8825
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.08
IIS Server: DL3
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8826 - 9050
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.09
IIS Server: DL4
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9051 - 10550
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.10
IIS Server: DL4
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 10551 - 12050
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.11
IIS Server: DL4
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12051 - 12275
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.12
IIS Server: DL5
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12276 - 13775
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.13
IIS Server: DL5
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13776 - 15275
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.14
IIS Server: DL5
SQL Server: via:sqlorca
Database: tpcc

User: sa
Protocol: HTML
w_id Range: 15276 - 15500
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.15
IIS Server: DL6
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15501 - 17000
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.16
IIS Server: DL6
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17001 - 18500
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.17
IIS Server: DL6
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18501 - 18725
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

IIS Server: DL7
Driver Engine: DRIVER.18
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18726 - 20225
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.19
IIS Server: DL7
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 20226 - 21725
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.20
IIS Server: DL7
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21726 - 21950
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.21
IIS Server: DL8
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21951 - 23450
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No
Driver Engine: DRIVER.22
IIS Server: DL8
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23451 - 24950
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.23
IIS Server: DL8
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24951 - 25175
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.24
IIS Server: DL9
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 25176 - 26675
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.25
IIS Server: DL9
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26676 - 28175
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal

User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.26
IIS Server: DL9
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28176 - 28400
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.27
IIS Server: DL10
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28401 - 29900
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.28
IIS Server: DL10
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29901 - 31400
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.29
IIS Server: DL10
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31401 - 31625
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.30
IIS Server: DL11
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31626 - 33125
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.31
IIS Server: DL11
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33126 - 34625
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.32
IIS Server: DL11
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34626 - 34850
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.33
IIS Server: DL12
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34851 - 36350
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.34
IIS Server: DL12
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36351 - 37850
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.35
IIS Server: DL12
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37851 - 38075
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.36
IIS Server: DL13
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38076 - 39575
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.37
IIS Server: DL13

SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 39576 - 41075
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.38
IIS Server: DL13
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41076 - 41300
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.39
IIS Server: DL14
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 41301 - 42800
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.40
IIS Server: DL14
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 42801 - 44300
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No
Driver Engine: DRIVER.41
IIS Server: DL14
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 44301 - 44525
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.42
IIS Server: DL15
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 44526 - 46025
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.43
IIS Server: DL15
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 46026 - 47525
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.44
IIS Server: DL15
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 47526 - 47750
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
District id: 1
User Count: 2250
Scale Down: No

Driver Engine: DRIVER.45
IIS Server: DL16
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 47751 - 49250
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.46
IIS Server: DL16
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 49251 - 50750
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.47
IIS Server: DL16
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 50751 - 50975
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.48
IIS Server: DL17
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 50976 - 52475
w_id Min Warehouse: 1

w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.49
IIS Server: DL17
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 52476 - 53975
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.50
IIS Server: DL17
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 53976 - 54200
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.51
IIS Server: DL18
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 54201 - 55700
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.52
IIS Server: DL18
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 55701 - 57200
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.53
IIS Server: DL18
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 57201 - 57425
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.54
IIS Server: DL19
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 57426 - 58925
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.55
IIS Server: DL19
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 58926 - 60425
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.56
IIS Server: DL19
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 60426 - 60650
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.57
IIS Server: DL20
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 60651 - 62150
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.58
IIS Server: DL20
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 62151 - 63650
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.59
IIS Server: DL20
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 63651 - 63875
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.60
IIS Server: DL21
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 63876 - 65375
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.61
IIS Server: DL21
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 65376 - 66875
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.62
IIS Server: DL21
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 66876 - 67100
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.63
IIS Server: DL22
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 67101 - 68600
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.64
IIS Server: DL22
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 68601 - 70100
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.65
IIS Server: DL22
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 70101 - 70325
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.66
IIS Server: DL23
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 70326 - 71825
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.67
IIS Server: DL23
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 71826 - 73325
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.68
IIS Server: DL23
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 73326 - 73550
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.69
IIS Server: DL24
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 73551 - 75050
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.70
IIS Server: DL24
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 75051 - 76550
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.71
IIS Server: DL24
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 76551 - 76775
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.72
IIS Server: DL25
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 76776 - 78275
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.73
IIS Server: DL25
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 78276 - 79775
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.74
IIS Server: DL25
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 79776 - 80000
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.75
IIS Server: DL26
Database: tpcc
SQL Server: via:sqlorca
User: sa
Protocol: HTML
w_id Range: 80001 - 81500
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.76
IIS Server: DL26
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 81501 - 83000
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.77
IIS Server: DL26
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 83001 - 83225
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.78
IIS Server: DL27
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 83226 - 84725
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

IIS Server: DL27
Driver Engine: DRIVER.79
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 84726 - 86225
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.80
IIS Server: DL27
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 86226 - 86450
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.81
IIS Server: DL28
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 86451 - 87950
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.82
IIS Server: DL28
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 87951 - 89450
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1

Scale Down: No

Driver Engine: DRIVER.83
IIS Server: DL28
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 89451 - 89675
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.84
IIS Server: DL29
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 89676 - 91175
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.85
IIS Server: DL29
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 91176 - 92675
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.86
IIS Server: DL29
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 92676 - 92900
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.87
IIS Server: DL30
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 92901 - 94400
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.88
IIS Server: DL30
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 94401 - 95900
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.89
IIS Server: DL30
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 95901 - 96125
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Driver Engine: DRIVER.90
IIS Server: DL31
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 96126 - 97625
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.91
IIS Server: DL31
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 97626 - 99125
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 15000
District id: 1
Scale Down: No

Driver Engine: DRIVER.92
IIS Server: DL31
SQL Server: via:sqlorca
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 99126 - 99350
w_id Min Warehouse: 1
w_id Max Warehouse: 99350
Scale: Normal
User Count: 2250
District id: 1
Scale Down: No

Number of Parameter Sets: 3

AuditShutdown

Menu		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Delay
	New Order	44.72	200.00	200.00	0.10	5.00	0.10
	Payment	43.07	200.00	200.00	0.10	5.00	0.10
	Delivery	4.07	200.00	200.00	0.10	5.00	0.10
	Stock Level	4.07	200.00	200.00	0.10	20.00	0.10

		Order Status	4.07	200.00	200.00	
	0.10	5.00	0.10			

Audit

			Txn	Think	Key	RT	RT	Menu
			Weight	Time	Time	Delay	Fence	Delay
		New Order	44.96	12.05	18.00			
	0.10	5.00	0.10					
		Payment	43.01	12.05	3.00			
	0.10	5.00	0.10					
		Delivery	4.01	5.05	3.00			
	0.10	5.00	0.10					
		Stock Level	4.01	5.05	3.00			
	0.10	20.00	0.10					
		Order Status	4.01	10.05	3.00			
	0.10	5.00	0.10					

~Default

			Txn	Think	Key	RT	RT	Menu
			Weight	Time	Time	Delay	Fence	Delay
		New Order	10.00	10.05	2.01			
	0.10	5.00	0.10					
		Payment	10.00	10.05	2.01			
	0.10	5.00	0.10					
		Delivery	1.00	10.05	2.01			
	0.10	5.00	0.10					
		Stock Level	1.00	10.05	2.01			
	0.10	5.00	0.10					
		Order Status	1.00	10.05	2.01			
	0.10	5.00	0.10					

Appendix D 60 Day Space Requirements

TPC-C 60 Day Space Requirements						
Warehouses	112000				TpmC	1,231,433.77
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	112000	12104	648	638		13390
District	1120000	124560	768	6,266		131594
Customer	3360000000	2443636552	152463408	129,804,998		2725904958
History	3360000000	200597288	749664		34,222,495	201346952
NewOrder	1008000000	17960192	41248	900,072		18901512
Orders	3360000000	109714520	246504		108,345,582	109961024
OrderLine	33599976544	2203277400	5189832		387,631,645	2208467232
Item	100000	9416	520	497		10433
Stock	11200000000	3584000000	7552912	179,577,646		3771130558
Total		8,559,332,032	166,245,504	310,290,116	530,199,722	9,035,867,652
	MB					
Dynamic Space	2,454,677	Sum of Data for Order, Orderline and History				
Static Space	6,369,413	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	431,825	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	32,278,888					
60 Day Space GB	31,522.35	GB				
Log Size	1,425,000.00	MB				
KB Per New Order	6.03	KB				
8 hr log MB	3,480,572	MB				
8 hr log GB	3,398.9958	GB				
		Disks	Disks	Formatted Size	Space	
Space Usage	GB Needed	Measured	Size	Size	Available	
180 Day Space DB	31,522.35	592	18GB	16.955	10037.24	
		1094	36GB	33.915	37102.46	
			4GB	3.999	0.00	
Total DB		1686.00			**47,139.70**	
8-hr log + mirror	6,797.99	56	146GB	136.710	**7,655.76**	
OS, Swap	1	4	18GB	16.955	67.82	
		2	36GB	33.915	67.83	135.65
Total Storage	38,321.34	GB			54,863.28	

Appendix E 3rd Party Pricing

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

November 21, 2005

Hewlett-Packard Company
Gunter Zink
One Microsoft Way
Redmond, WA 98052

Mr. Zink:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars ($).

Part Number	Description	Unit Price	Quantity	Price
810-03134	**SQL Server 2005 Enterprise Itanium Edition** *Per Processor License* *Discount Schedule: Open Program - Level C* *Unit Price reflects a 6% discount from the retail unit price of $24,999.*	$23,432	64	$1,499,648
P73-00295	**Windows Server 2003 Standard Edition** *Server License Only - No CALs* *Discount Schedule: Open Program - No Level* *Unit Price reflects a 28% discount from the retail unit price of $999.*	$719	31	$22,289
254-00170	**Visual C++ Standard Edition** *No Discounts Applied*	$109	1	$109
N/A	**Microsoft Problem Resolution Services** *Professional Support* *(1 Incident)*	$245	1	$245

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by November 7, 2005.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at $245 per call.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

KVMs.com

USB Extenders

Free Shipping Everyday

We'll Beat Any Price

Brands | KVM Switch | Extenders | KVM over IP | KVM over CAT5 | KVM Cables | Video Switch | Video Splitter | Remote Reboot | KVM Drawer | Console Switch | Computer Cables

Order Tracking : Shopping Cart : Customer Care Center : Technical Help Center

Search

Delete	Accessories	No.	Product	Quantity	Price	Extended
	Add-On	5150	Startech StarView 1 X 32 KVM Switch w/OSD	1	$ 1599.00	1,599.00
	Add-On	5250	KVM Premium 6ft PS2 3-in-1 Cable Kit	31	$ 9.00	279
				Sub Total		**1,878.00**

Check-Out

XpressTrack

Enter XpressTrack Code (if any)

GO

PromoCode

Enter promotional Code (if any)

GO

Contact

Toll-Free:
1-800-KVM-Switch
(1-800-586-7948)
Intl/local:
1.770.971.1924
Fax:
1.770.971.2639
Email:
sales@kvms.com
Mail:
PO Box 669875
Marietta, GA 30066-0115

Chat Live click here

Instructions

- To increase the quantity, just select the quantity in the quantity box.

  Quantity 1

- If you like to remove an item, click on the Trash Can next to the item
- Please review your shopping cart.
- Once ready, click Check-Out to place your order.
- If you like to order by phone, Call us Toll-

SAN Marketplace

Where the storage professionals shop!™

1-866-4

Call u

Home | About Us | Support | Search | Checkout | Contact Us

Tuesday November 22nd 2005

Search GO

Shopping cart

3 Product(s) in cart

3 Prime-SB5200

9 SB5200-16A

52 Finisar FTRJ-8519 2Gb SFP

Total $47,688.96

> Checkout

Call Us Toll-Free

1-866-463-3372

Switches

Details	Name	Unit Price	Quantity	Total	Select
Prime-SB5200	Prime-SB5200	$1,327.32	9	$11,945.88	
SB5200-16A	SB5200-16A	$4,371.00	3	$13,113.00	
FTRJ8519P1BNL	Finisar FTRJ-8519 2Gb SFP	$84.00	52	$4,368.00	
		Grand Total:		$29,426.88	Delete
	Continue Shopping		Update Totals		

Press checkout to enter your customer and shipping information.

checkout

Brocade 3250 Switches Brocade 3252 Switches Brocade 3200 Series Software
Brocade 3850 Switches Brocade 3852 Switches Brocade 3800 Series Software Brocade 3800 Series Parts
Brocade 3900 Switches Brocade 3900 Series Software Brocade 3900 Series Parts
DS8B2 / DS8B3 DS8Bx Software DS16B2 / DS16B3 DS16Bx Software
DS24M2 DS24M2 Software
DS32B2 DS32B2 Software
DS32M2 DS32M2 Software
Sphereon 4300 Sphereon 4500 Sphereon 3232 IPS-3300
SANbox 5200 / 5202 Switches SANbox 5200 License Keys SANbox2-64 SANbox2-16

Host Bus Adapters

AMCC/JNI

Hewlett Packard Company	HP Integrity Superdome Enterprise Server			TPCC Rev 5.5 October 2005 Report Date November 29 2005		
Description	Price Key	Part Numbr	Unit Price	0	Extended Price	3 Yr Maint Price
Superdome left cabinet	1	A5201A, Opt. 429	$236,716	1	$236,716	
Superdome right cabinet	1	A5202A, Opt. 429	$251,200	1	$251,200	
IPF Superdome Cell Board (sx1000)	1	A6866A	$19,200	16	$307,200	
3 Year Svc & Support Price (Hardware and Software)						$888,659
HP Superdome Itanium2 1.6 GHz 9M 2 CPU	1	AD003A	$41,800	32	$1,337,600	
Superdome 128GB Memory Pack	1	AB547A	$282,880	8	$2,263,040	
PCI-x I/O chassis	1	A6864A	$16,805	8	$134,440	
Core I/O card	1	A6865A	$1,045	1	$1,045	
Gig E - TX Adapter	1	A7061A	$270	1	$270	
Graphics USB Card	1	A6869A	$299	1	$299	
FC-HBA 2GB, 1 Channel (LP1050)	1	AB467A	$1,520	4	$6,080	
Qlogic QLA-2350 Fibre-Channel VI Adapter	1	A390975	$1,256	16	$20,096	
HP Smart Array Controller 6402 (boot)	1	A9890A	$1,669	1	$1,669	
HP Smart Array Controller 6404 (incl boot)	1	A9891A	$2,549	30	$76,470	
36GB, 15krpm Ultra3 Wide disk (Incl Boot)	1	286776-B22	$299	1686	$504,114	
36GB, 15krpm Ultra3 Wide disk (10% Spares)	1	286776-B22	$299	169	$50,531	
Storageworks MSA30 SB (incl boot)	1	302969-B21	$2,829	121	$342,309	
Storageworks MSA30 SB (10% Spares)	1	302969-B21	$2,829	13	$36,777	
HP9000 Standard Rack System E41	1	A4902A	$1,910	14	$26,740	
HP Power Distribution Unit 120-240V	1	A5137AZ	$145	27	$3,915	
Modular Storage Array 1000	1	201723-B22	$6,995	4	$27,980	
MSA1000 controller	1	218231-B22	$4,290	4	$17,160	
Storage MSA1000 Support 3yrs 24x7 4hr incl disks	1	HA110A3 #6FG	$3,222	4		$12,888
146GB, 10krpm disk	1	286716-B22	$499	56	$27,944	
5M LC to LC Cable Kit	1	221692-B22	$82	4	$328	
TA5300 Enclosure for DAT tape	1	C7508AZ	$729	1	$729	
DVD Rom drive	1	Q1592A	$649	1	$649	
DAT Tape	1	C7497B	$1,049	1	$1,049	
sx1000 Superdome SMS, rack	1	A9802B	$6,500	1	$6,500	
	Server Subtotal				**$5,682,850**	**$901,547**
Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1	1	T2372A opt064	$128,000	1	$128,000	
	Server Software Subtotal				**$128,000**	**$0**
DL140 Dual 3.06GHz Xeon 1GB 80GB ATA	1	356533-B21	$1,799	31	$55,769	
HP 3 Yr. 4h 24x7 ProLiant DL140 Hardware Support	1	U9510E	$419	12		$5,028
1GB PC2100 ECC DDR	1	351109-B21	$419	31	$12,989	
HP Mouse	1	P5304M	$28	1	$28	
HP Enhanced Keyboard (USB/PS2)	1	DC852A#ABA	$29	1	$29	
S7540 17in CRT Monitor	1	PF997AA	$149	1	$149	
Qlogic QLA-2350 FC VI Adapter (incl 10% spares)	1	A390975	$1,256	35	$43,960	
5M LC to LC Cable Kit	1	221692-B22	$82	48	$3,936	
	Client Subtotal				**$116,860**	**$5,028**
HP's Large Configuration Discount *	*** Discounts:**				**-$2,098,024**	**-$358,544**
	Total:				**$3,829,686**	**$548,031**
Price Key: 1-HP	**3 year cost of ownership:**					**$4,377,717**

* A 35.94% discount was based on the overall value of the specific components from HP (Price Key 1) in this single quotation. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.