



---

# TPC Benchmark® C Full Disclosure Report

---

HP Integrity Superdome  
using Microsoft SQL Server 2005 Enterprise Edition 64-bit  
on Microsoft Windows Server 2003, Datacenter Edition (64-bit)  
SP1

First Edition  
June 7, 2005

First Edition - June 7, 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., June 7, 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 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	Microsoft Windows Server 2003, Datacenter Edition (64-bit) SP1
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$5,820,285 USD	1,082,203 tpmC	\$5.38 USD per tpmC	Dec 7, 2005

## Auditor

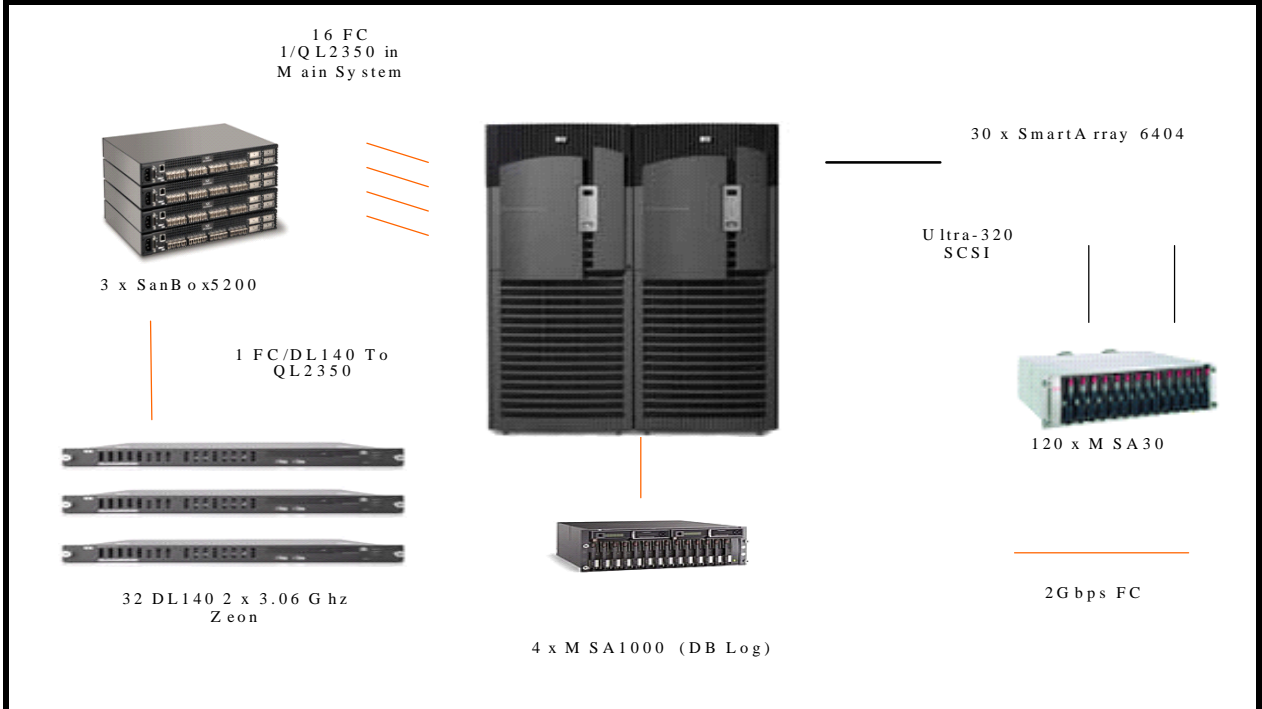
The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per tpmC®, were audited by Lorna Livingtree of Performance Metrics to verify compliance with the relevant TPC specifications.




# HP Integrity Superdome

TPC-C Revision  
5.4

<b>Total System Cost</b>	<b>TPC Throughput</b>	<b>Price/Performance</b>	<b>Report Date</b>	<b>Availability Date</b>
\$5,820,285 USD	1,082,203 tpmC	\$5.38 USD per tpmC	June 7, 2005	Dec 7, 2005
<b>Processors</b>	<b>DataBase Manager</b>	<b>Operating System</b>	<b>Other Software</b>	<b>Number of Users</b>
64 Intel Itanium2 1.6 GHz 9M - Server 64 x Intel Xeon 3.06GHz - Client	Microsoft SQL Server 2005 Enterprise Edition 64-bit	Microsoft Windows Server 2003, Datacenter Edition (64- bit) SP1	Microsoft Visual C++ Microsoft COM+ Transaction Monitor	919,910



System Components	Server		Each Client	
	Qty	Type	Qty	Type
<b>Processors</b>	64	Intel Itanium 2 9M CPUs at 1.6GHz	2	3.06 GHz Intel Xeon
<b>Cache Memory</b>		9 MB L3 cache		512kbyte L2 Cache
<b>Memory</b>	512	2 Gbyte	2	1048 Mbytes
<b>Disk Controllers</b>	30	HP Smart Array 6404	1	ATA
	1	HP Smart Array 6402 (Boot)		
<b>Disk Drives</b>	1064	HP 36GB 15KRPM	1	80 Gbyte disk
	616	HP 18GB 15KRPM		
	56	HP 146GB, 10krpm U320		
	2	HP 36 GB 15KRPM drive for Boot		
	2	HP 18GB 15KRPM for Utility		
<b>Total Storage</b>		53743.4 Gbyte		80 Gbyte
<b>Tape Drives</b>	1	HP TA5300		
<b>Terminals</b>	1	Console Terminal	1	Console Terminal

 <b>Hewlett Packard Company</b>	<b>HP Integrity Superdome Enterprise Server</b>		TPC-C Rev. 5.0			
			Report Date: April, 2003			
Description	Price Key	Part Numbr	Unit Price	Qty	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 (LP9802)	1	AB232A	\$1,550	4	\$6,200	
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	1682	\$502,918	
Storageworks MSA30 SB (incl boot)	1	302969-B21	\$2,829	121	\$342,309	
Storage MSA30 Support 3yrs 24x7 4 hr incl disks	1	HA110A3 #7GT	\$1,827	121		\$221,067
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 MAS1000 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	
<b>Server Subtotal</b>					<b>\$5,594,466</b>	<b>\$1,122,614</b>
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	2	810-00846	\$19,909	64	\$1,274,176	\$245
<b>Server Software Subtotal</b>					<b>\$1,402,176</b>	<b>\$245</b>
DL140 Dual 3.06GHz Xeon 1GB 80GB ATA	1	356533-B21	\$1,799	32	\$57,568	
HP 3 Yr. 4h 24x7 ProLiant DL140 Hardware Support	1	U9510E	\$419	12		\$5,028
1GB PC2100 ECC DDR	1	351109-B21	\$419	32	\$13,408	
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	\$12	32	\$384	
Qlogic QLA-2350 FC VI Adapter (incl 10% spares)	1	A390975	\$1,256	37	\$46,472	
5M LC to LC Cable Kit	1	221692-B22	\$82	48	\$3,936	
<b>Client Subtotal</b>					<b>\$123,573</b>	<b>\$5,028</b>
Microsoft Windows 2003 Server	2	P73-00295	\$719	32	\$23,008	
Microsoft Visual C++ Standard	2	254-00170	\$109	1	\$109	
Microsoft Problem Resolution	2		\$245	1		\$245
<b>Client Software Subtotal</b>					<b>\$23,117</b>	<b>\$245</b>
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	53	\$4,452	
<b>Connectivity Subtotal</b>					<b>\$17,565</b>	<b>\$11,943</b>
HP's Large Configuration Discount *	<b>* Discounts:</b>				<b>-\$2,082,351</b>	<b>-\$398,336</b>
<b>Total:</b>					<b>\$5,078,546</b>	<b>\$741,739</b>
Price Key: 1-HP, 2-Microsoft, 3-SANMarketplace.com 4 - KVMs.com	<b>3 year cost of ownership:</b>					<b>\$5,820,285</b>
	tpmC:					1,082,000
* All discounts are based on US list prices and for similar quantities and configurations	<b>\$/tpmC:</b>					<b>\$5.38</b>
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.						

## Numerical Quantities Summary for HP Integrity Superdome

**MQTH, Computed Maximum Qualified Throughput**

**1082203 tpmC**

### Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	2.56s	9.82s	1.60s
Payment	2.52s	9.81s	1.57s
Order-Status	2.53s	9.82s	1.58s
Delivery (interactive portion)	0.82s	8.48s	0.32s
Delivery (deferred portion)	0.30s	3.22s	0.17s
Stock-Level	2.60s	10.04s	1.64s
Menu	0.85s	8.51s	0.33s

Response time delay added for emulated components 0.1 seconds

### Transaction Mix, in percent of total transactions

New-Order	44.93%
Payment	43.04%
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	18.06s	0.00s	12.05s	120.52s
Payment	3.00s	3.01s	3.06s	0.00s	12.05s	120.51s
Order-Status	2.00s	2.01s	2.05s	0.00s	10.05s	100.51s
Delivery (interactive)	2.00s	2.01s	2.05s	0.00s	5.06s	50.5s
Stock-Level	2.00s	2.01s	2.05s	0.00s	5.05s	50.5s

### Test Duration

Ramp up time	88 minutes
Measurement interval	120 minutes
Transactions during measurement interval	300617278
Ramp down time	5 minutes

### Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	25 minutes

# Table of Contents

Abstract.....	1
Overview.....	1
TPC Benchmark® C Metrics .....	1
Standard and Executive Summary Statements .....	1
Auditor.....	1
<b>Table of Contents.....</b>	<b>5</b>
Preface.....	7
Document Structure .....	7
TPC Benchmark® C Overview.....	7
System Overview.....	8
General Items.....	9
Test Sponsor .....	9
Application Code and Definition Statements.....	9
Parameter Settings .....	9
Configuration Diagrams.....	9
Chapter 1 Logical Database Design.....	11
1.1 Table Definitions .....	11
1.2 Physical Organization of the Database.....	11
1.3 Insert and Delete Operations.....	11
1.4 Partitioning.....	11
1.5 Replication, Duplication or Additions.....	11
Chapter 2 Transaction and Terminal Profiles.....	12
2.1 Random Number Generation .....	12
2.2 Input/Output Screen Layout.....	12
2.3 Priced Terminal Feature Verification.....	12
2.4 Transaction Statistics.....	12
2.5 Presentation Manager or Intelligent Terminal .....	13
2.6 Queuing Mechanism .....	13
Chapter 3 Transaction and System Properties.....	14
3.1 Transaction System Properties (ACID Tests).....	14
3.2 Atomicity Tests .....	14
3.2.1 COMMIT Transaction.....	14
3.2.2 ROLLBACK Transaction .....	14
3.3 Consistency Tests .....	14
3.4 Isolation Tests .....	15
3.5 Durability Tests.....	15
3.5.1 Loss of Data / Loss of Log.....	15
3.5.2 Loss of System / Memory .....	16
Chapter 4 Scaling and Database Population.....	17
4.1 Database Layout .....	17
4.2 Initial Cardinality of Tables .....	20
4.3 60 Day Space .....	21
4.3.1 Transaction Log Space Requirements.....	21
4.4 Type of Database Used.....	21
4.5 Database Mapping.....	22
Chapter 5 Performance Metrics and Response Time.....	23
5.1 Throughput .....	23
5.2 Response Times .....	23
5.3 Keying and Think Times .....	23
5.4 Response Time Frequency .....	24
5.4.1 New Order Response Time.....	24
5.4.2 Payment Response Time Distribution.....	25

5.4.3	Order Status Response Time .....	26
5.4.4	Delivery Response Time Distribution.....	27
5.4.5	Stock Level Response Time .....	28
5.4.6	Response Time Versus Throughput.....	29
5.4.7	New Order Think Time Distribution .....	30
5.4.8	Throughput Versus Time Distribution .....	31
5.5	Steady State Determination.....	31
5.6	Work Performed During Steady State.....	31
5.6.1	Checkpoint.....	32
5.6.2	Checkpoint Conditions .....	32
5.6.3	Checkpoint Implementation.....	32
5.7	Measurement Period Duration .....	32
5.8	Regulation of Transaction Mix .....	32
5.9	Transaction Mix.....	32
5.10	Transaction Statistics.....	33
5.11	Checkpoint Count and Location.....	33
Chapter 6	SUT, Driver and Communications Definition .....	34
6.1	RTE Description .....	34
6.2	Emulated Components .....	34
6.3	Functional Diagram .....	34
6.4	Networks .....	34
6.5	Operator Intervention .....	34
Chapter 7	Pricing.....	35
7.1	System Pricing.....	35
7.2	General Availability, Throughput and Price Performance .....	35
7.3	Country Specific Pricing.....	35
7.4	Usage Pricing.....	35
Chapter 8	Audit .....	36
8.1	Auditor's Information .....	36
Appendix A	Source Code.....	39
Appendix B	Database Load .....	145
B.1	Database Options.....	177
B.2	Table definitions .....	178
B.3	Stored Procedures.....	183
Appendix C	Tunable Parameters .....	376
C.1	Microsoft SQL Server 8.0 Configuration Parameters.....	444
C.2	Client System Configuration Parameters .....	445
C.3	RTE Input Parameters.....	462
Appendix D	60 Day Space Requirements.....	476
Appendix E	3 <sup>rd</sup> Party Pricing .....	477



# 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. It meets the requirements of the TPC Benchmark® C Standard Specification, Revision 5.4 dated April 2005. TPC Benchmark® C was developed by the Transaction Processing Performance Council (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 On Line Transaction Processing (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.

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 36GB hard disk pair (mirrored RAID 1), attached to an HP SmartArray 6402 controller. 1 additional mirrored pair of 18GB drives (36 GB priced) were used for utility storage of scripts, the build environment, etc.

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 9802 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. 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 bandwidth between the web-clients and the database server was 2GBps. 32 remote terminal emulators (RTEs) emulated 919,910 users executing the standard TPC-C workload. Each web-client had two embedded Gigabit LAN adapters, one of which 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 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 15 SoftNuma nodes of 4 CPUs each, with a 16<sup>th</sup> SoftNuma node of 3 CPUs. The load was reduced accordingly on this last node. A 17<sup>th</sup> SoftNuma node was configured with the just the last CPU. This CPU/SofNuma node was used for the checkpoint process. A script doing continuous checkpoints of 1500 seconds (25 minutes) was started on one of the web clients after steady state was reached with a different connection port than the regular webclient to database server traffic. 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 16 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.

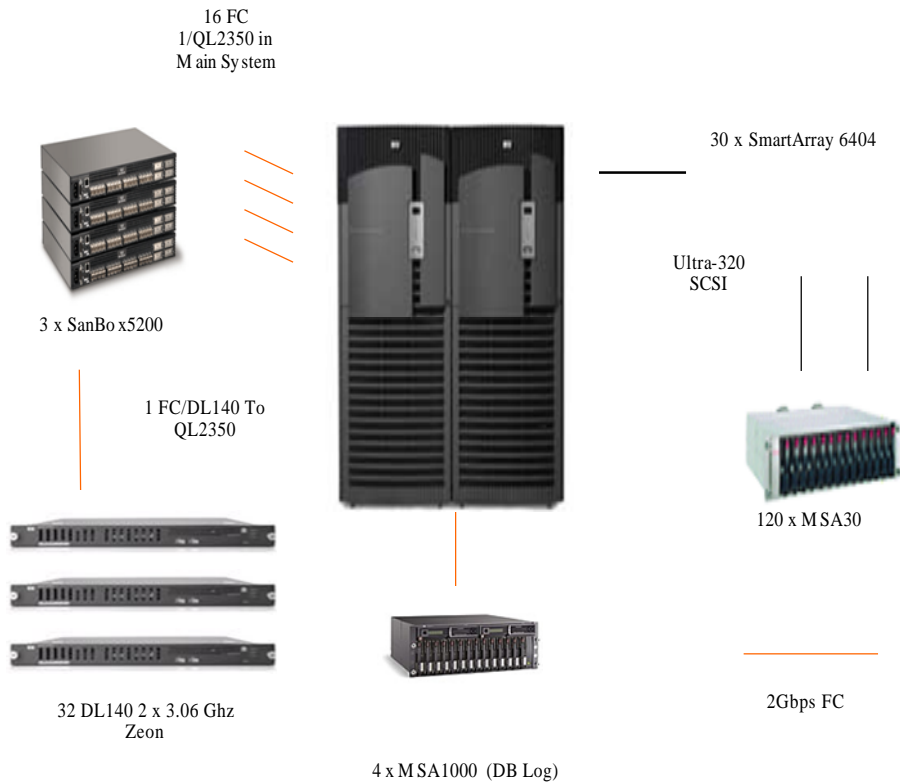
Appendix D contains the 60 day space calculations.

### 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, containing 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	60.07%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.93%
	Payment	43.04%
	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 3.4.2.1 to 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 91991 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 91991 (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 91,991 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 919,910 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. 919,910 users were logged in to the database and ran transactions.
4. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
5. After another 5 minutes, power to the system was cut off.
6. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
7. The RTE was stopped.
8. The server machine was powered on again and rebooted.
9. Microsoft SQL Server was restarted and performed an automatic recovery.
10. A new count of D\_NEXT\_O\_ID was taken.
11. This number was compared with the number of new orders reported by the RTE.
12. 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 9802 FC HBAs.

The measured database configuration used a total of 1738 disks, which included 1064 HP 36GB 15KRPM and 616 18GB 15KRPM drives for data, 56 HP 146GB, 10krpm U320 drives for log, and one 36GB mirrored pair for the operating system. 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 2 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).

**Table 2: 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 94.24 GB	G:\MNT\MISC1 Raw 54.23 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				
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 94.24B	G:\MNT\MISC2 Raw 54.23GB
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				
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\BACKUP\2\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 94.24GB	G:\MNT\MISC60 Raw 54.23GB
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				
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		

## 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**

<b>Table</b>	<b>Occurrences</b>
Warehouse	98,000
District	980,000
Customer	2,940,000,000
History	2,940,000,000
Orders	2,940,000,000
New Orders	882,000,000
Order Line	840,041,944
Stock	9,800,000,000
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/I, 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 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 94236 MB each and misc\_fg consist of 60 partitions at 52227 MB each as shown in the createdb.sql. The log was configured with 20,000 MB at database creation, and was expanded to 1,125,000 MB after data base creation and load.



## Chapter 5 Performance Metrics and Response Time

### 5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 1,082,203  
Price per TpmC®: \$5.38 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	1.60s	2.56s	9.82s
Payment	1.57s	2.52s	9.81s
Order-Status	1.58s	2.53s	9.82s
Delivery (interactive portion)	0.32s	0.82s	8.48s
Delivery (deferred portion)	0.17s	0.30s	3.22s
Stock-Level	1.64s	2.60s	10.04s
Menu	0.33s	0.85s	8.51s

### 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	18.06s
Payment	3.00	3.01s	3.06s
Order Status	2.00	2.01s	2.05s
Interactive Delivery	2.00	2.01s	2.05s
Stock Level	2.00	2.01s	2.05s

**Table 6: Transaction Think Times**

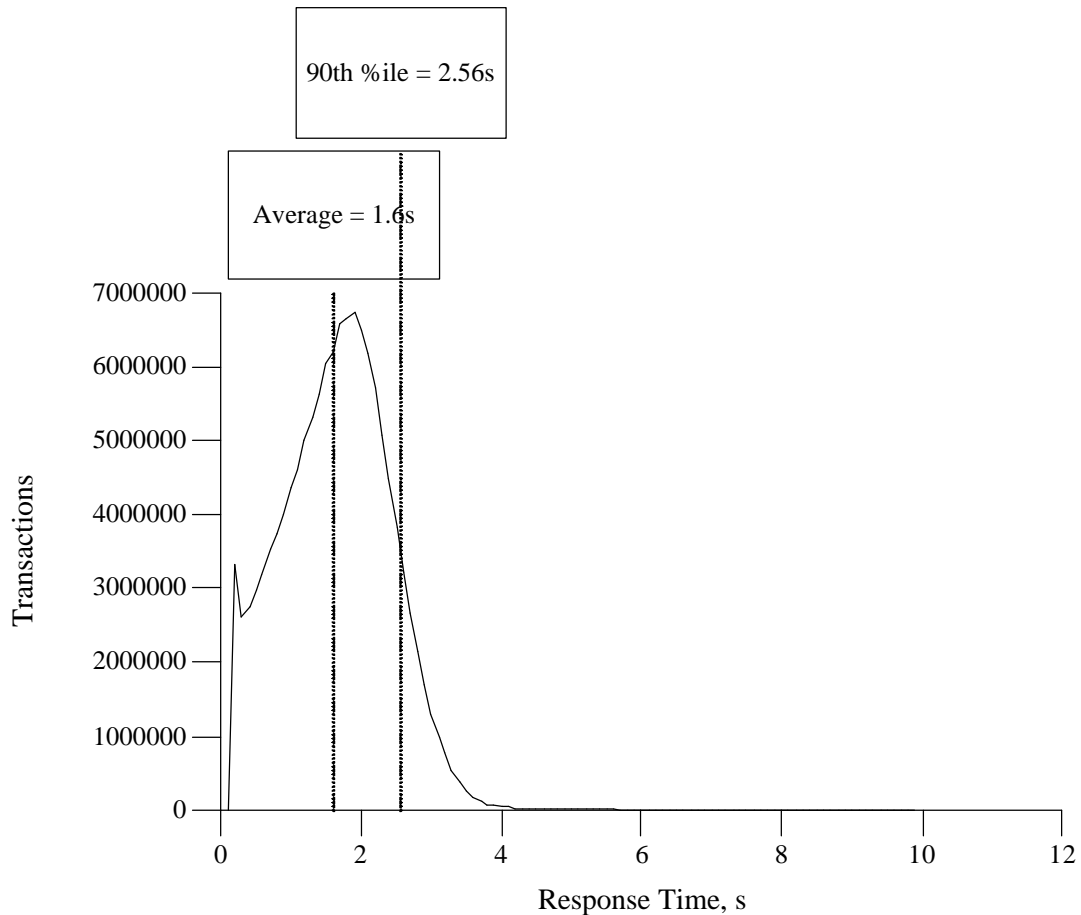
Think Times	Minimum	Average	Maximum
New Order	0	12.05s	120.52s
Payment	0	12.05s	120.51s
Order Status	0	10.05s	100.51s
Interactive Delivery	0	5.06s	50.5s
Stock Level	0	5.05s	50.5s

## 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 Clause 5.6.4) 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

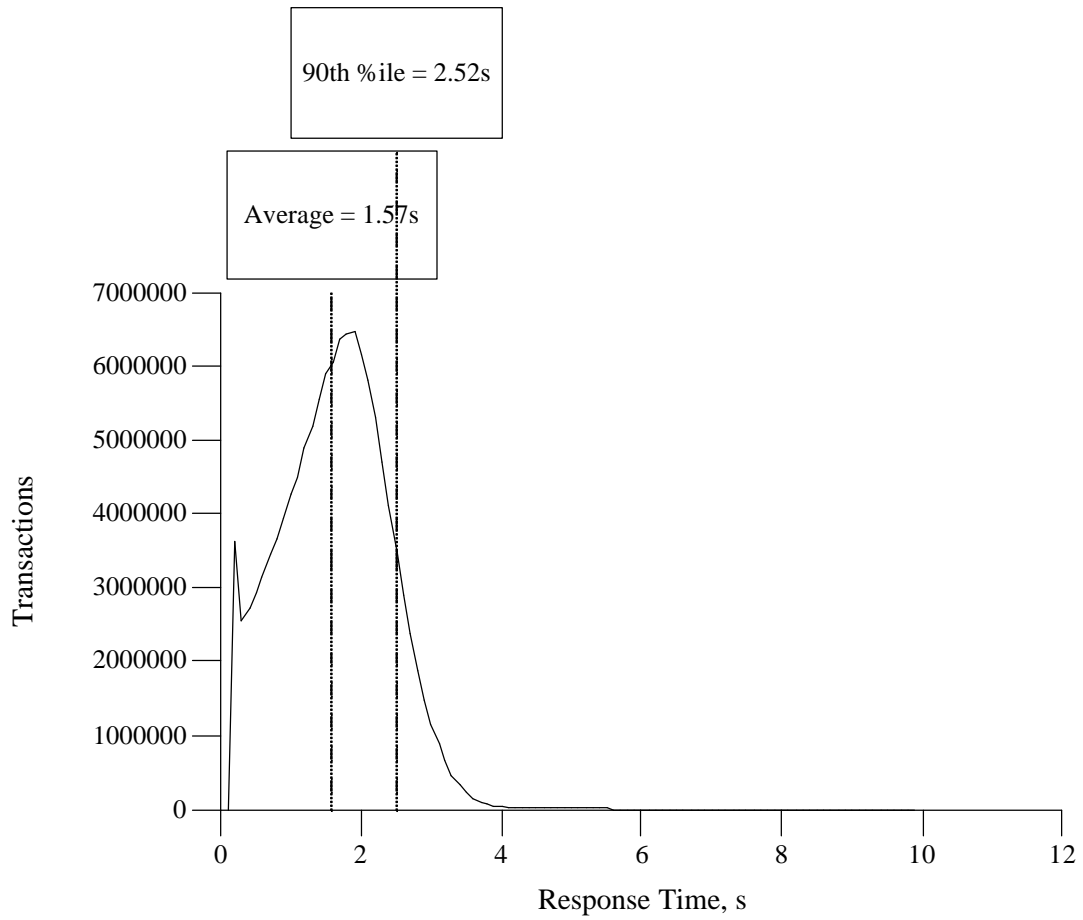
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

## 5.4.2 Payment Response Time Distribution

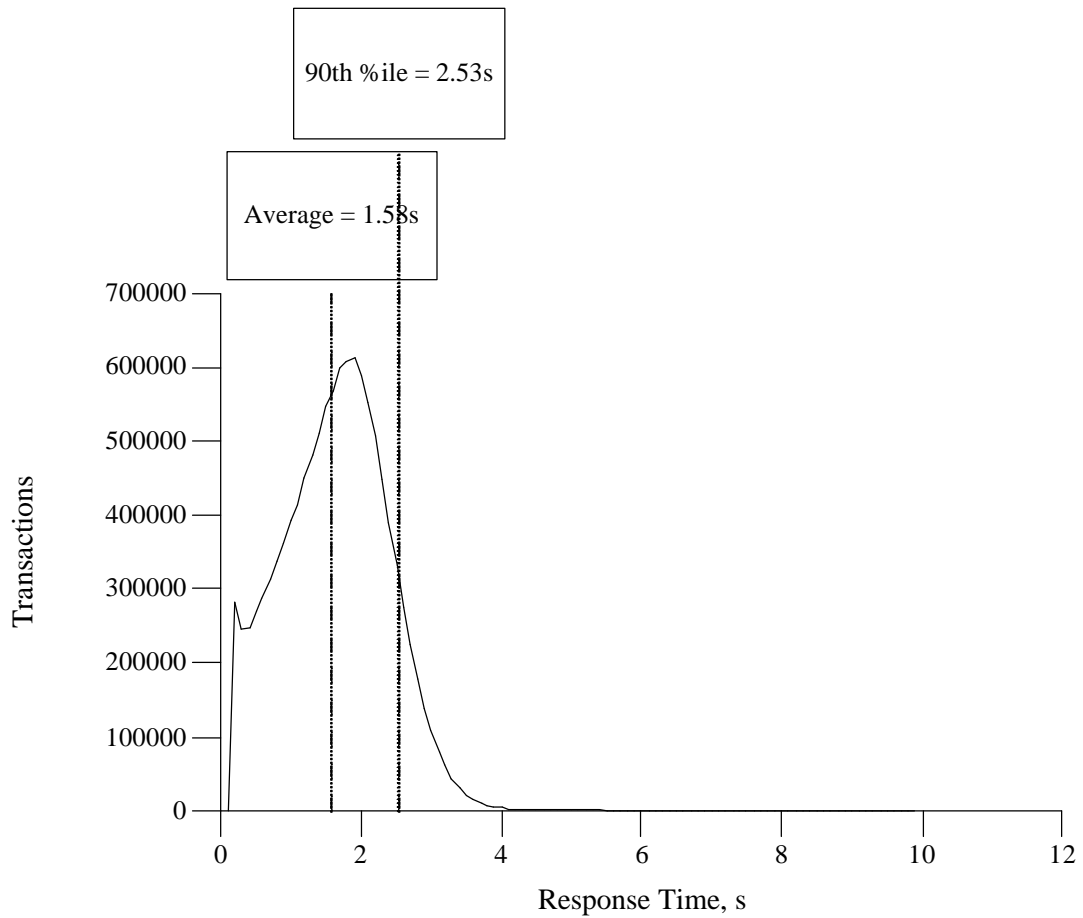
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

### 5.4.3 Order Status Response Time

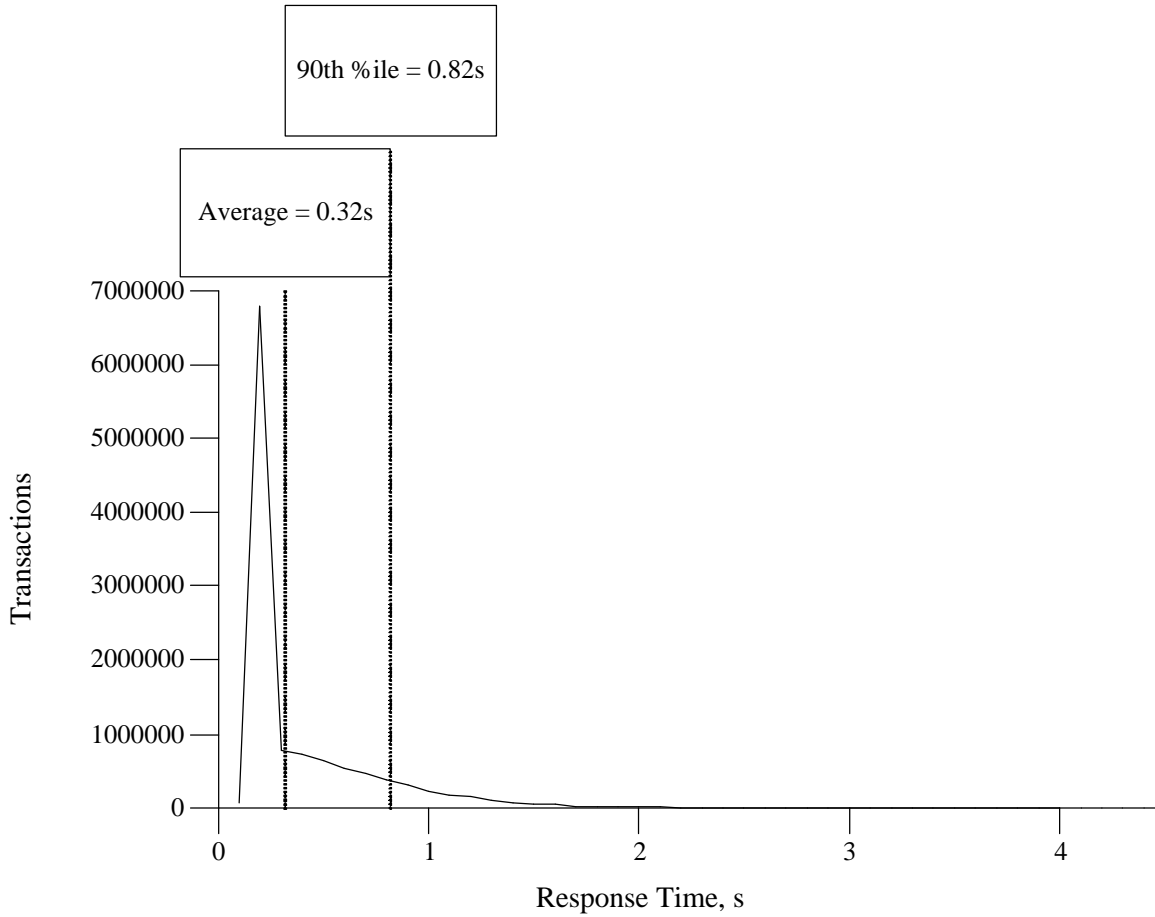
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

## 5.4.4 Delivery Response Time Distribution

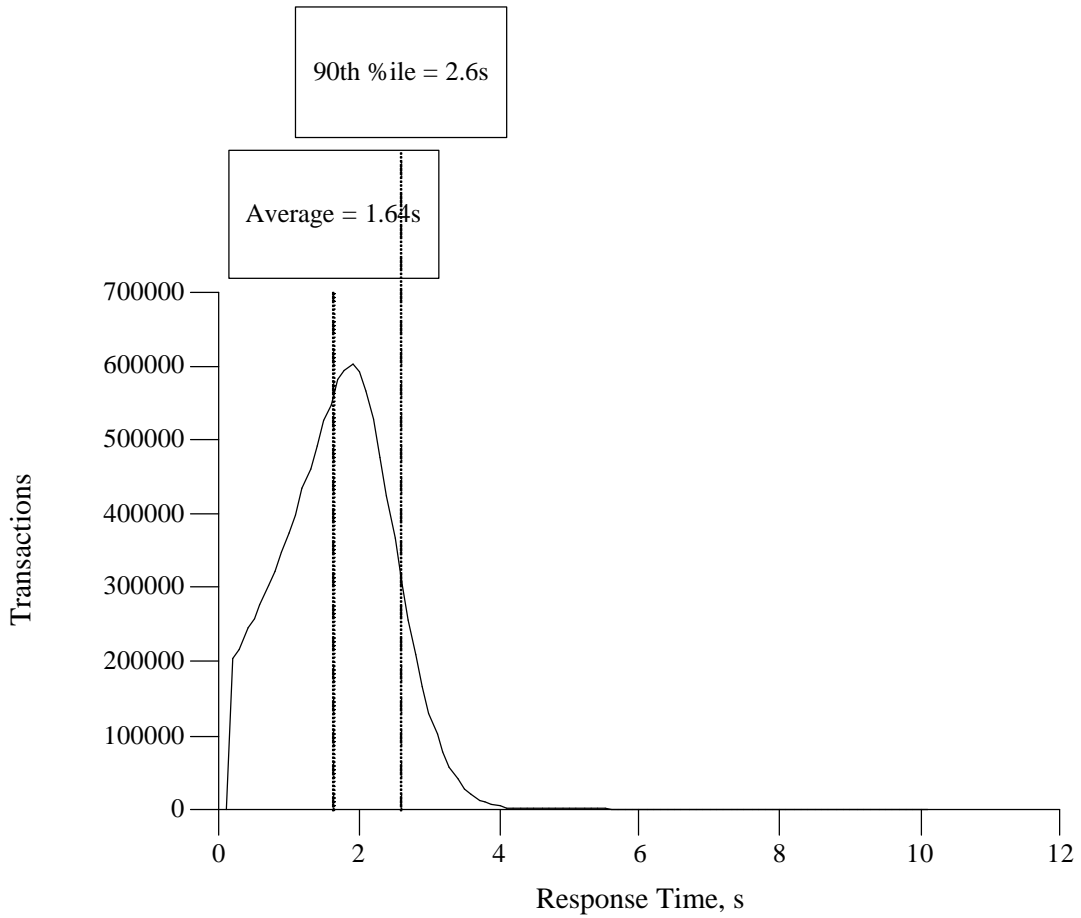
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

### 5.4.5 Stock Level Response Time

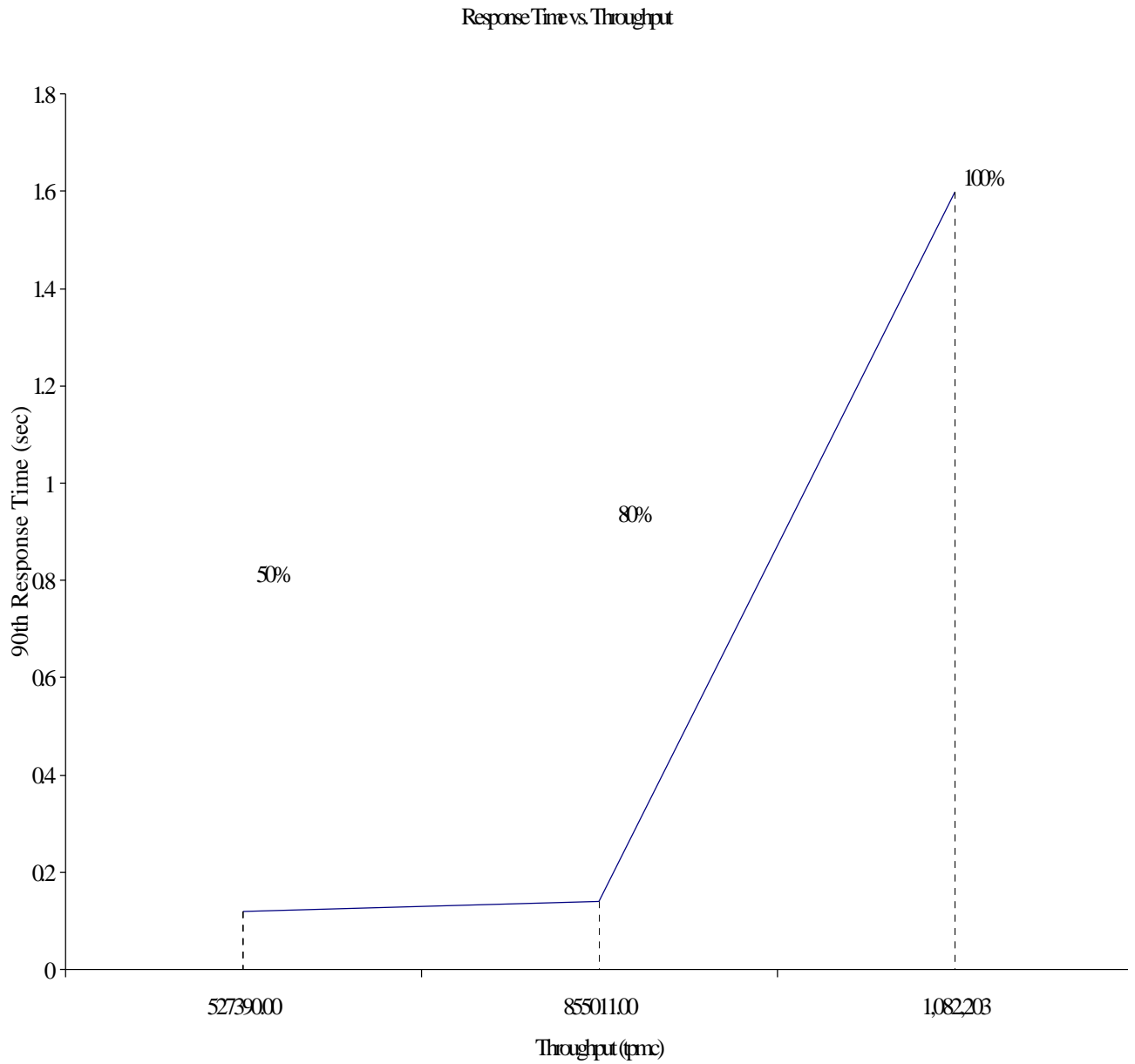
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

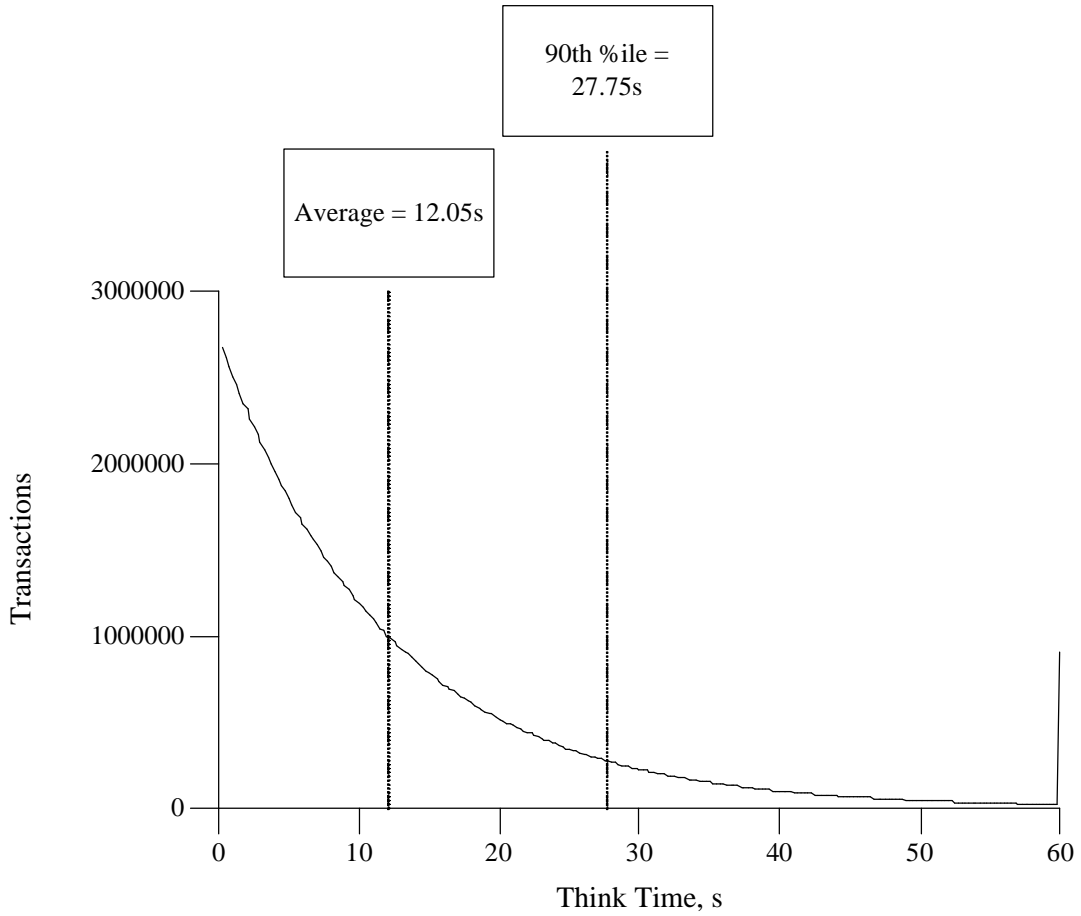
## 5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

Figure 9: New Order Think Time Distribution

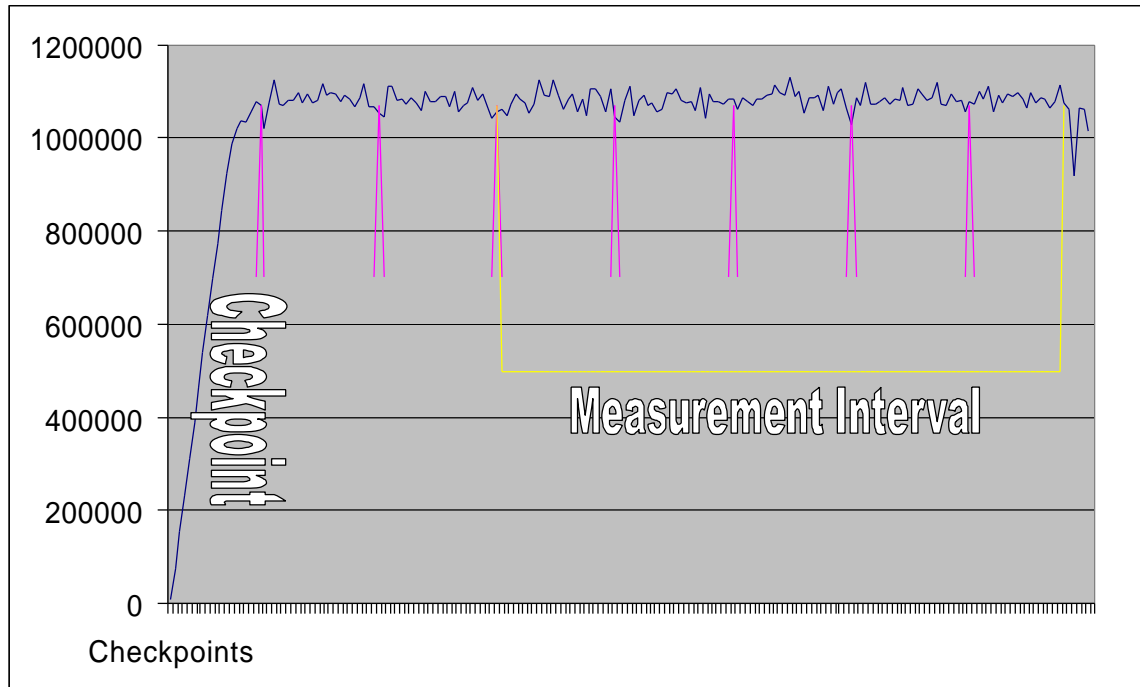


Think time frequency distribution for New Order transaction



## 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 1500" syntax in Microsoft SQL Server 2005 Enterprise Edition 64-bit was used to force the checkpoints to an interval of 25 minutes. The checkpoints were affinitized to a 1 processor SoftNuma node 16 using connection affinity. One of the web clients whose network connections were affinitized to that cell was used to issue the checkpoints. This was done with the same QL2350 VI card that was used for network traffic. 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 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 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	44.93%
Payment	43.04%
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 8 checkpoints within the measurement interval and 4 checkpoint before the measurement interval.

**Table 8: Measurement Interval and Checkpoints**

Event	From	To
Measured Interval	3:02:27	5:02:27
Checkpoint	3:02:27	3:27:27
Checkpoint	3:27:27	3:52:27
Checkpoint	3:52:27	4:17:28
Checkpoint	4:17:28	4:42:28

## 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**

<b>Maximum qualified throughput:</b>	1,082,203 tpmC
<b>Price per tpmC:</b>	\$5.38 USD per tpmC
<b>Availability:</b>	Dec 7, 2005

### 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 per-processor licenses.
- Microsoft Windows Server 2003, Datacenter edition (64-bit)
- 32 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:



June 2, 2005

MR. JASON GOERTZ  
HP WIE Performance Redmond  
Hewlett-Packard Company  
14475 NE 24<sup>th</sup> 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	1,023 GB	616 @ 18 GB 1064 @ 36 GB 56 @ 146 GB	2.56	<b>1,082,202</b>
32 Clients: ProLiant DL140 each with:				
2 Pentium Xeon @ 3.06 Ghz	Main: 2 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 98,000 warehouses, of which 91,991 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 919,910 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:

(1) 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.

(2) The performance run was found to have 3 transactions out of 300,617,278 that did not meet the minimum key time. The key times were 17.999, 17.998 and 17.994 seconds and should have been 18 seconds. These were positioned such that no 2 hour window could avoid them. Considering the magnitude of the transaction volume, these transactions were insignificant to measured throughput, in my opinion. Therefore, this performance run was accepted as an ease of benchmarking measure.

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
 *
 *      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
    int d_id; //district id
    int iSyncId; //synchronization id
    int iTickCount; //time of last access;

    CTMCC_BASE *pTxn;
```

```
} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational interface for terminal
//id support
typedef struct _TERM
{
    int iNumEntries; //total allocated
    int iFreeList;
    int iMasterSyncId; //next available terminal array element or -1 if none
    CLIENTDATA *pClientData; //synchronization id
    //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);
void GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR
NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
void TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int
iErrorNum, int iTermId, int iSyncId, char *szErrorText, char *szForm);
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)
#ifndef _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

#ifdef !_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

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

#endif 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 <stdlib.h>
#include <malloc.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 definitions 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: DIIMain
 *
 * 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

*
*      ul_reason_for_call      reason for call      DWORD
*
*      lpReserved              reserved for        LPVOID
future use
*
* RETURNS:        BOOL      FALSE
                  errors occurred 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
bxn 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() );
}
}

```

```

(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->ErrorNum(), e->ErrorText());
>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);
lstrcpy(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.
*
* ARGUMENTS: 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 be 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,
break;
NULL, INPUT_FORM, szBuffer);
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(txnDeliRec != 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++;

(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",
                >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 relevant 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
GetIntKeyValue(&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);
        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)
    sprintf( 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              char
                 key
value to look for
*
                 *pValue          char
                 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
if 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              char
                 key
value to look for
*
                 WEBERROR
                 error value to throw if key not
found
*
                 WEBERROR
                 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)

```

```

    {
        // New Term is available
        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 )
{
    sprintf(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=\"TERMINID\" 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)
{
    sprintf(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=\"TERMINID\" 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 = sprintf(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=\"TERMINID\" 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></PRE><HR>"
              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
              "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
              "</FORM></HTML>" );
    }
    else

```

```

    {
        wprintf(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> <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 = wprintf(szForm,

```

```

" <HTML><HEAD><TITLE>TPC-C New
Order</TITLE></HEAD><BODY>
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="\TERMINID\" 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 += wprintf(szForm+c, "Warehouse:
%6.6d ", Term.pClientData[iTermId].w_id);

        strcpy( szForm+c,
            "District: <INPUT
Date: <BR>"
            "Customer: <INPUT
Credit:
"
            "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 += wprintf(szForm+c, "Warehouse:
%6.6d District: %2.2d
Date: ",
pNewOrderData->w_id,
pNewOrderData->d_id);

        if ( bValid )
        {
            c += wprintf(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 += sprintf(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 += sprintf(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 committed. Total: $%8.2f
",
                pNewOrderData-
>total_amount);
        else
            c += sprintf(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 = sprintf(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=\"TERMINID\" 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 += sprintf(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 += sprintf(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 += sprintf(szForm+c,
            "<BR> <BR>Warehouse:
%6.6d District: %2.2d<BR>"
            "%-20s %-
20s<BR>"
            "%-20s %-
20s<BR>"
            "%-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 += sprintf(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 += sprintf(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 );
            strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");
            strcpy(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..\">"
            );
        }
    }
    /* 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 = sprintf(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=\TERMIID\ VALUE=\%d">"
" <INPUT TYPE=\hidden"
NAME=\SYNCID\ VALUE=\%d">"
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:
NAME=\CLT*\ SIZE=23><BR>"
"Order-Number: Entry-
"Order-Number: Entry-
"Supply-W Item-Id Qty
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> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
" <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>",
>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);
}
strcpy( 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=\TERMIID\ 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();

```

```

    pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
    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          client browser
                  lpszQueryString
http command string
*
                  *pPaymentData  PAYMENT_DATA
                  payment data structure
                  pointer to
*/

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 );

        _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 );

        _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) ) 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;
}

```

#### 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 szDBNames[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 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 ( !strcmp(szTmp, szDBNames[ODBC]) )
        pReg->eDB_Protocol = ODBC;
    else if ( !strcmp(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 ( !strcmp(szTmp,
szTxnMonNames[TUXEDO]) )
        pReg->eTxnMon = TUXEDO;
    else if ( !strcmp(szTmp,
szTxnMonNames[ENCINA]) )
        pReg->eTxnMon = ENCINA;
    else if ( !strcmp(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 ( !strcmp(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 ( RegQueryValueEx(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
    //dlib 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 dlib 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
//Error from TPCW during receive
#define ERR_TYPE_TPCC_CONN       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_TPCC_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
#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 immediately 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());

    ErrorText();

    MessageBox(hwnd, szTmp, m_szApp,
MB_OK);
}

```

```

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
ERR_TYPE_OS;};
    char*
    char*
    int
(int)m_eAction; }
    void
        Action
        m_eAction;

private:
    char m_szMsg[ERR_MSG_BUF_SIZE];
};

class CMemoryErr : public CBaseErr
{
public:
    CMemoryErr();

    int
ERR_TYPE_MEMORY;};
    char*
    }
    char*
};

class CBufferOverflowErr : public CBaseErr
{
public:
    CBufferOverflowErr(int,LPTSTR);

    int
ERR_BUF_OVERFLOW;};

```

```

        char*
OVERFLOW";};
        char*
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
return ERR_TYPE_XML_PROFILE;};
    virtual char
PROFILE"; };
    virtual char
    *ErrorText();
    virtual int
return m_eCode; };

    int
ErrorAction() { return (int)m_eAction; }
//virtual void Draw(HWND hwnd, LPCTSTR
szStr = NULL)
};

```

```

        //
        //};
        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
                // TPC-W error type from tpcc ole-db txn module
#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 immediately 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

```

```

};
        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

```

```

        LoadFromXML,
        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.
#ifdef 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 () =
virtual void Delivery () =
virtual void StockLevel () =
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
 * Microsoft Corp.
 */

#ifdef _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
memory
 * held for a very short time. When a lock is taken a
semaphore
 * location is exchanged. All other threads that want this
 * lock wait by spinning and sometimes sleeping on a
 * 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 m_Spinlock;
    Semaphore;
    volatile LONG m_Spinlock;
    volatile LONG Waiting;

    #ifdef _DEBUG
        // Counters for debugging
        volatile LONG TotalLocks;
        volatile LONG TotalSleeps;
        volatile LONG TotalSpins;
        volatile LONG TotalWaits;
    #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 server. 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 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
    BYTE      TxnSubType;
// = 0
// 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
#define TPCW record          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
}

```

```

//
// file
//
// RecMap[RecMapSize];
// 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 IFilePointer;
//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 ISavePtFilePointer;
int iSavePtNextRec;
JULIAN_TIME lastTTS;
//when writing
sorted output, used to verify records are sorted
BOOL bWrite;
//writing log file
BOOL bCrashOpen;
// 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 if the 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 be 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; //ptr to current buffer
BYTE *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);
        int Sort(void);
        PTXN_RECORD_HEADER GetSortedRecord();
        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 <sqlldb.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_dlib.h"
#define DEFCLPACKSIZE
4096
// version string; must match return value from tpcc_version stored
proc
```

```
const char sVersion[] = "4.10.000";
const iMaxRetries = 10;
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 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          destination string pointer
                  *pDest
 *
                  char          source string pointer
                  *pSrc
 *
                  int           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_erno == 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 (dbprocmshandle(login, msg_handler) == NULL)
        ThrowError(CDBLIBERR::eDbProcHandler);

    DBSETUSER(login, szUser);
    DBSETLPWD(login, szPassword);
    DBSETHOST(login, szHost);
    DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
    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) != SUCCEEDED)
    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); //
            @threshold 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);
        }
    }
}

```

```

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;
        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;
    DBDATETIME    edatetime;
    DBDATERECD    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_bxn.NewOrder.w_id);
        dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_bxn.NewOrder.d_id);
        dbrpcparam(m_dbproc, NULL,
0, SQLINT4, -1, -1, (BYTE *) &m_bxn.NewOrder.c_id);
        dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_bxn.NewOrder.o_ol_cnt);

// check whether any order
lines are for a remote warehouse
        m_bxn.NewOrder.o_all_local =
1;
        for (i = 0; i <
m_bxn.NewOrder.o_ol_cnt; i++)
        {
                if
(m_bxn.NewOrder.OL[i].ol_supply_w_id != m_bxn.NewOrder.w_id)
                {
                        m_bxn.NewOrder.o_all_local = 0; // at least one remote
warehouse

                        break;
                }
        }
        dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_bxn.NewOrder.o_all_local);

        for (i = 0; i <
m_bxn.NewOrder.o_ol_cnt; i++)
        {
                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_bxn.NewOrder.OL[i].ol_i_id);

                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE
*) &m_bxn.NewOrder.OL[i].ol_supply_w_id);

                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE
*) &m_bxn.NewOrder.OL[i].ol_quantity);
        }

        if (dbrpcexec(m_dbproc) ==
FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

```

```

        m_bxn.NewOrder.total_amount
= 0;
        for (i = 0;
i < m_bxn.NewOrder.o_ol_cnt; i++)
        {
                if
(dbresults(m_dbproc) != SUCCEED)

                        ThrowError(CDBLIBERR::eDbResults);

                if
(dbnumcols(m_dbproc) != 5)

                        ThrowError(CDBLIBERR::eWrongNumCols);

                if
(dbnxtrow(m_dbproc) != REG_ROW)

                        ThrowError(CDBLIBERR::eDbNextRow);

                if(pData=dbdata(m_dbproc, 1))

                        UtilStrCpy(m_bxn.NewOrder.OL[i].ol_i_name, pData,
dbdatlen(m_dbproc, 1));

                if(pData=dbdata(m_dbproc, 2))

                        m_bxn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *)
pData);

                if(pData=dbdata(m_dbproc, 3))

                        UtilStrCpy(m_bxn.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_bxn.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_bxn.NewOrder.OL[i].ol_amount,
8);

                m_bxn.NewOrder.total_amount =
m_bxn.NewOrder.total_amount + m_bxn.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 (dbnxtrow(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_bxn.NewOrder.w_tax, 8);

                if (pData=dbdata(m_dbproc,
2))

                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,2), SQLFLT8, (BYTE *)&m_bxn.NewOrder.d_tax, 8);

                if (pData=dbdata(m_dbproc,
3))

                        m_bxn.NewOrder.o_id = (*(DBINT *) pData);

                if (pData=dbdata(m_dbproc,
4))

                        UtilStrCpy(m_bxn.NewOrder.c_last, pData,
dbdatlen(m_dbproc, 4));

```

```

5)          if (pData=dbdata(m_dbproc,
                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);
6)          if (pData=dbdata(m_dbproc,
                UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));
7)          if (pData=dbdata(m_dbproc,
                {
                    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,
2)          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 &&
>m_msgtext, sErrTimeoutExpired) != NULL)) &&
                {
                    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()
{
    DBDATETIME datetime;
    DBDATERECDaterec;
    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.b_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);
2)          if (pData=dbdata(m_dbproc,
                UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
3)          if (pData=dbdata(m_dbproc,
                {
                    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_street_1, 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_bxn.Payment.c_id == 0)
            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
            m_bxn.Payment.exec_status_code = eOK;

        return;
    }
    catch (CSQLERR *e)
    {
        if ((e->m_msgno == 1205 ||
            == iErrOleDbProvider &&
            >m_msgtext, sErrTimeoutExpired) != NULL) &&
            iMaxRetries))
        {
            // hit deadlock;
            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
    DBDATETIME datetime;
    DBDATAREC daterec;

    int
    RETCODE rc;
    const BYTE *pData;

    i;
    iTryCount = 0;

```

```

        ResetError();
        while (TRUE)
        {
            try
            {
                dbrpcinit(m_dbproc,
                "tpcc_orderstatus", 0);

                dbrpcparam(m_dbproc, NULL,
                0, SQLINT4, -1, -1, (BYTE *) &m_bxn.OrderStatus.w_id);
                dbrpcparam(m_dbproc, NULL,
                0, SQLINT1, -1, -1, (BYTE *) &m_bxn.OrderStatus.d_id);
                dbrpcparam(m_dbproc, NULL,
                0, SQLINT4, -1, -1, (BYTE *) &m_bxn.OrderStatus.c_id);

                // if customer id is zero, then
                order status is by name
                if (m_bxn.OrderStatus.c_id ==
                0)
                    dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1,
                    strlen(m_bxn.OrderStatus.c_last), (unsigned char
                    *)m_bxn.OrderStatus.c_last);

                if (dbrpcexec(m_dbproc) ==
                FAIL)
                    ThrowError(CDBLIBERR::eDbRpcExec);

                // Get order lines
                if (dbresults(m_dbproc) !=
                SUCCEEDED)
                    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 =
                if (rc ==
                NO_MORE_ROWS)
                    break;
                if (rc !=
                REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);

                if(pData=dbdata(m_dbproc, 1))
                    m_bxn.OrderStatus.OL[i].ol_supply_w_id =
                (*(DBSMALLINT *) pData);
                if(pData=dbdata(m_dbproc, 2))
                    m_bxn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
                if(pData=dbdata(m_dbproc, 3))
                    m_bxn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *)
                pData);
                if(pData=dbdata(m_dbproc, 4))
                    dbconvert(m_dbproc, SQLNUMERIC, (LPBYTE)pData,
                    dbdatlen(m_dbproc,4),
                    SQLFLT8, (BYTE
                    *)&m_bxn.OrderStatus.OL[i].ol_amount, 8);
                if(pData=dbdata(m_dbproc, 5))
                    {
                        datetime = *((DBDATETIME *) pData);
                        dbdatecrack(m_dbproc, &daterec, &datetime);
                        m_bxn.OrderStatus.OL[i].ol_delivery_d.year =
                    daterec.year;
                        m_bxn.OrderStatus.OL[i].ol_delivery_d.month =
                    daterec.month;
                        m_bxn.OrderStatus.OL[i].ol_delivery_d.day =
                    daterec.day;
                        m_bxn.OrderStatus.OL[i].ol_delivery_d.hour =
                    daterec.hour;
                    }

```

```

        m_bxn.OrderStatus.OL[i].ol_delivery_d.minute =
daterec.minute;

        m_bxn.OrderStatus.OL[i].ol_delivery_d.second =
daterec.second;
    }
    i++;
}
m_bxn.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_bxn.OrderStatus.c_id = (*(DBINT *) pData);
        if(pData=dbdata(m_dbproc,
2))
            UtilStrCpy(m_bxn.OrderStatus.c_last, pData,
dbdatlen(m_dbproc,2));
        if(pData=dbdata(m_dbproc,
3))
            UtilStrCpy(m_bxn.OrderStatus.c_first, pData,
dbdatlen(m_dbproc,3));
        if(pData=dbdata(m_dbproc,
4))
            UtilStrCpy(m_bxn.OrderStatus.c_middle, pData,
dbdatlen(m_dbproc, 4));
        if(pData=dbdata(m_dbproc,
5))
        {
            datetime =
*((DBDATETIME *) pData);

```

```

        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_bxn.OrderStatus.o_entry_d.year = daterec.year;

        m_bxn.OrderStatus.o_entry_d.month = daterec.month;

        m_bxn.OrderStatus.o_entry_d.day = daterec.day;

        m_bxn.OrderStatus.o_entry_d.hour = daterec.hour;

        m_bxn.OrderStatus.o_entry_d.minute = daterec.minute;

        m_bxn.OrderStatus.o_entry_d.second = daterec.second;
    }
    if(pData=dbdata(m_dbproc,
6))
        m_bxn.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_bxn.OrderStatus.c_balance, 8);
    if(pData=dbdata(m_dbproc,
8))
        m_bxn.OrderStatus.o_id = (*(DBINT *) pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        if (m_bxn.OrderStatus.o_ol_cnt
== 0)
            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
        else if (m_bxn.OrderStatus.c_id
== 0 && m_bxn.OrderStatus.c_last[0] == 0)
            throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
            m_bxn.OrderStatus.exec_status_code = eOK;

        return;
    }
    catch (CSQLERR *e)
    {

```

```

        if ((e->m_msgtext) != m_OLSQL)
        == iErrOleDbProvider &&
        >m_msgtext, sErrTimeoutExpired) != NULL)) &&
        iMaxRetries))
        {
            // hit deadlock;
            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_bxn.Delivery.w_id);
            dbrpcparam(m_dbproc, NULL,
0, SQLINT1, -1, -1, (BYTE *) &m_bxn.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 &&
                    >m_msgtext, sErrTimeoutExpired) != NULL)) &&

                    (iMaxRetries)
                    {
                        backoff for increasingly longer period

                        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

#ifdef 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.

```

```

#ifdef DBLIB
__declspec( dllimport )
#endif
#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_erno =
iErr; m_iTryCount = 0; };

        CTPCC_DBLIB_ERR( int iErr, int iTryCount ) {
m_erno = iErr; m_iTryCount = iTryCount; };

        int        m_erno;

        int        m_iTryCount;

        int        ErrorType()
    {return ERR_TYPE_TPCC_DBLIB;};
        char*      ErrorTypeStr() { return "TPCC
DBLIB"; }
        int        ErrorNum()
    {return m_erno;};

        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\_odbcc.dll    tpcc\_odbcc.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 <sqltext.h>

```

```

#define COMPILE_FOR_SNAC // define that to compile for
SQL Native Client; comment out to use MDAC
#ifdef COMPILE_FOR_SNAC
#include <odbccs.h>
#else
// Compile for SNAC
#include <sqlcli.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_odbcc.h"

// version string; must match return value from tpcc_version stored
proc

const char sVersion[] = "4.10.000";

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_erno == 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_descOrderStatusCols = 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;

#ifdef COMPILER_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             !NativeError;
    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, &!NativeError,
(BYTE *)&szMsg, sizeof(szMsg), NULL);
        if (rc == SQL_NO_DATA)
            break;

        // check for deadlock
        if (!NativeError == 1205 || (!NativeError ==
iErrOleDbProvider &&
sErrTimeoutExpired) != NULL))
            pODBCErr->m_bDeadLock =
TRUE;

        // capture the (first) database error
        if (pODBCErr->m_NativeError == 0 &&
!NativeError != 0)
            pODBCErr->m_NativeError =
!NativeError;

```

```

        // 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, CODBCERR::eBindCol);

        //Compose Stock Level statement
        _snprintf(m_szStockLevelCommand,
sizeof(m_szStockLevelCommand)/sizeof(m_szStockLevelCommand[0]),
L"%s %stmpcc_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_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]),
L"(call
%stppcc_neworder(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)",
m_szSPPrefix);

    m_iBeginNewOrderVariablePart = 29 +
wcslen(m_szSPPrefix); // fixed part + prefix part

////////////////////////////////////
////////////////////////////////////
//
//      Now initialize New Order that works on no
duplicate (w_id,i_id) pairs
//      and returns one result set for lineitem
details.
//
//      m_hstmt = m_hstmtNewOrderNoDuplicates;

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    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_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 row-wise binding
    if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.NewOrder.OL[0]), SQL_IS_UIINTEGER) !=
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_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_descNewOrderNoDuplicatesCols2,
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_szNewOrderNoDuplicatesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNewOrderNo
DuplicatesCommand[0]),
        L"{{call %stpcp_neworder_new
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"}

```

```

        L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"}",
m_szSPPrefix);
    m_iBeginNewOrderNoDuplicatesVariablePart = 33 +
wcslen(m_szSPPrefix); // fixed part + prefix part
}

//
// Returns true if there are duplicate (warehouse_id,
// item_id)
// lineitem pairs in New Order input parameters.
//
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
    int i, j;
    for (i = 0; i < m_txn.NewOrder.o_ol_cnt; ++i)
    {
        for (j = i+1; j < m_txn.NewOrder.o_ol_cnt;
++j)
        {
            if
(m_txn.NewOrder.OL[i].ol_i_id == m_txn.NewOrder.OL[j].ol_i_id)
            {
                return true;
            }
        }
    }
    return false;
}

void CTPCC_ODBC::NewOrder()
{
    if (m_bCallNoDuplicatesNewOrder)
    {
        if (DuplicatesInNewOrder())
        {
            NewOrderDuplicates();
        }
        else
        {
            NewOrderNoDuplicates();
        }
    }
    else
    {
        NewOrderDuplicates();
    }
}

```

```

void CTPCC_ODBC::NewOrderDuplicates()
{
    int
    i;
    RETCODE                                rc;
    int
    iTryCount = 0;

1    2                                // 0

012345678901234567890123456789        //
    wchar_t
    szSqlTemplate[iMAX_SP_NAME_LEN];

    tpcc_neworder(?,?,?,?,?"          // = L"{call

L"?,?,?,?,?,?,?,?,?,?,?,?,?"          //

L"?,?,?,?,?,?,?,?,?,?,?,?,?"          //

L"?,?,?,?,?,?,?,?,?,?,?,?,?);"        //

    m_hstmt = m_hstmtNewOrder;

    // associate the parameter and column bindings for this
transaction
    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols1, SQL_IS_POINTER
) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

    // clip statement buffer based on number of parameters
// fixed part is 29 chars and variable part is 6 chars per
line item
    wcsncpy(szSqlTemplate, m_szNewOrderCommand);
    i = m_iBeginNewOrderVariablePart +
m_bxn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L" )" );

    // check whether any order lines are for a remote
warehouse
    m_bxn.NewOrder.o_all_local = 1;
    for (i = 0; i < m_bxn.NewOrder.o_ol_cnt; i++)
    {
        if (m_bxn.NewOrder.OL[i].ol_supply_w_id !=
m_bxn.NewOrder.w_id)

```

```

        {
            m_bxn.NewOrder.o_all_local =
0; // at least one remote warehouse
            break;
        }
    }

    while (TRUE)
    {
        try
        {
            m_BindOffset = 0;
            rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
            if (rc != SQL_SUCCESS && rc
!= SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

                // Get order line results
                m_bxn.NewOrder.total_amount
= 0;

                for (i = 0;
i < m_bxn.NewOrder.o_ol_cnt; i++)
                {
                    // set the bind
                    m_BindOffset = i
* sizeof(m_bxn.NewOrder.OL[0]);

                    if (
SQLFetch(m_hstmt) == SQL_ERROR)
                        ThrowError(CODBCERR::eFetch);

                        // move to the
                        next resultset
                        if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
                            ThrowError(CODBCERR::eMoreResults);

                            m_bxn.NewOrder.total_amount +=
m_bxn.NewOrder.OL[i].ol_amount;
                }

                // 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);
                if ( SQLFetch(m_hstmt) ==
SQL_ERROR)
                    ThrowError(CODBCERR::eFetch);

                    SQLFreeStmt(m_hstmt,
SQL_CLOSE);

                    if (m_no_commit_flag == 1)
                    {
                        m_bxn.NewOrder.total_amount *= ((1 +
m_bxn.NewOrder.w_tax + m_bxn.NewOrder.d_tax) * (1 -
m_bxn.NewOrder.c_discount));

                        m_bxn.NewOrder.exec_status_code = eOK;
                    }
                    else
                        m_bxn.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;

1   2   3                               // 0

0123456789012345678901234567890123    //
    wchar_t
    szSqlTemplate[iMAX_SP_NAME_LEN];

tpcc_neworder_new(?,?,?,?)           // = L"{call

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
    wcsncpy(szSqlTemplate,
m_szNewOrderNoDuplicatesCommand);
    i = m_iBeginNewOrderNoDuplicatesVariablePart +
m_bxn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L"");

// check whether any order lines are for a remote
warehouse
    m_bxn.NewOrder.o_all_local = 1;
    for (i = 0; i < m_bxn.NewOrder.o_ol_cnt; i++)
    {
        if (m_bxn.NewOrder.OL[i].ol_supply_w_id !=
m_bxn.NewOrder.w_id)
        {
            m_bxn.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_bxn.NewOrder.total_amount
= 0;
        for (i = 0;
i < m_bxn.NewOrder.o_ol_cnt; i++)
        {
            m_bxn.NewOrder.total_amount +=
m_bxn.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_bxn.NewOrder.total_amount *= ((1 +
m_bxn.NewOrder.w_tax + m_bxn.NewOrder.d_tax) * (1 -
m_bxn.NewOrder.c_discount));

            m_bxn.NewOrder.exec_status_code = eOK;
        }
        else
            m_bxn.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_bxn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_bxn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
&m_bxn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_bxn.Payment.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_bxn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_bxn.Payment.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_bxn.Payment.c_last), 0, &m_bxn.Payment.c_last,
sizeof(m_bxn.Payment.c_last), NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_bxn.Payment.c_id,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_last,
sizeof(m_bxn.Payment.c_last),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_bxn.Payment.h_date,
0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.w_street_1,
sizeof(m_bxn.Payment.w_street_1), NULL) !=
SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.w_street_2,
sizeof(m_bxn.Payment.w_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.w_city,
sizeof(m_bxn.Payment.w_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.w_state,
sizeof(m_bxn.Payment.w_state), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.w_zip,
sizeof(m_bxn.Payment.w_zip),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.d_street_1,
sizeof(m_bxn.Payment.d_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.d_street_2,
sizeof(m_bxn.Payment.d_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.d_city,
sizeof(m_bxn.Payment.d_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.d_state,
sizeof(m_bxn.Payment.d_state), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.d_zip,
sizeof(m_bxn.Payment.d_zip),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_first,
sizeof(m_bxn.Payment.c_first),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_middle,
sizeof(m_bxn.Payment.c_middle), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_street_1,
sizeof(m_bxn.Payment.c_street_1), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_street_2,
sizeof(m_bxn.Payment.c_street_2), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_city,
sizeof(m_bxn.Payment.c_city),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_state,
sizeof(m_bxn.Payment.c_state),
NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_zip,
sizeof(m_bxn.Payment.c_zip),
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_phone,
sizeof(m_bxn.Payment.c_phone), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_bxn.Payment.c_since,
0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_credit, sizeof(m_bxn.Payment.c_credit), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_bxn.Payment.c_credit_lim, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_bxn.Payment.c_discount, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_bxn.Payment.c_balance, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_bxn.Payment.c_data,
sizeof(m_bxn.Payment.c_data),
NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);

    //Compose Payment statement
    _snwprintf(m_szPaymentCommand,
sizeof(m_szPaymentCommand)/sizeof(m_szPaymentCommand[0]),
L"%s{call %stppc_payment (?,?,?,?,?,?)}",
m_szSPPrefix);
}

void CTPCC_ODBC::Payment()
{
    RETCODE rc;
    int iTryCount = 0;

    m_hstmt = m_hstmtPayment;

    if (m_bxn.Payment.c_id != 0)
        m_bxn.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);

        int i = 0;
        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 %stppc_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
    _snprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryCommand[0]),
L"%s{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_bxn.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 tpc\_c\_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.
#ifdef 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, //
        eConnOption,
        // error from SQLSetConnectOption
        eConnect,
        // error from SQLConnect
        eAllocStmnt,
        // error from SQLAllocStmnt
        eExecDirect, //
        eBindParam,
        // error from SQLBindParameter
        eBindCol,
        // error from SQLBindCol
        eFetch,
        // error from SQLFetch
        eFetchScroll, //
        eMoreResults,
        // error from SQLMoreResults
        ePrepare,
        // error from SQLPrepare
        eExecute,
        // error from SQLExecute
        eSetEnvAttr, //
        eSetStmntAttr //
    };

    error from SQLSetEnvAttr
    error from SQLSetStmntAttr

};

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
    {return ERR_TYPE_ODBC;};

    char*
    {return "ODBC"; };

    int
    {return m_NativeError;};

    char*
    {return m_odbcerrstr;};

    int
    {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_erno =
    iErr; m_iTryCount = 0; };

    CTPCC_ODBC_ERR( int iErr, int iTryCount ) {
    m_erno = iErr; m_iTryCount = iTryCount; };

    int m_erno;
    int m_iTryCount;

    int
    {return ERR_TYPE_TPCC_ODBC;};

    char*
    {return "TPCC
    ODBC"; };
};

```

```

        int                ErrorNum();
    {return m_erno;};

    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;    // beginning of the
variable part in NewOrder statement
    wchar_t
m_iBeginNewOrderNoDuplicatesVariablePart;    //
beginning 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
    SQLINTEGER                m_BindOffset;
    SQLINTEGER                m_RowsFetched;
    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( LPCWSTR 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 PORORDER_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
( LPCWSTR 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.
#ifdef 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;

// 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 iError;
{
    NEW_ORDER_DATA      NewOrder;
    PAYMENT_DATA        Payment;
    DELIVERY_DATA       Delivery;
    STOCK_LEVEL_DATA    StockLevel;
    ORDER_STATUS_DATA   OrderStatus;
} u;
} *m_pTxn;

public:
    VARIANT m_vTxn;
    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);
```

### 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 definitions 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_G0TAB0CADDR_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);

```

```

private: // helper methods
    BOOL m_bCanBePooled;
    CTPCC_BASE *m_pTxn;

    struct COM_DATA
    {
        int retval;
        int error;
        union
        {
            NEW_ORDER_DATA
            PAYMENT_DATA
            DELIVERY_DATA
            STOCK_LEVEL_DATA
            ORDER_STATUS_DATA
        };
    };

};

////////////////////////////////////
// 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()

```

```

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
#ifdef 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 <atbase.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 <atcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atimpl.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
definitions 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 );

        DBLIB)

            if (Reg.eDB_Protocol ==
            {
                strcpy(
                szDllName, Reg.szPath );

                strcat(
                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 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, hCallNoDuplicatesNewOrder);
            else if (Reg.eDB_Protocol == ODBC)
                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
    {
        >pvData;
        pData = (COM_DATA*)txn_in.parray-
        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);
        >pvData;
        pData = (COM_DATA*) txn_out->parray-
        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
    {
        >pvData;
        pData = (COM_DATA*)txn_in.parray-
        >BuffAddr_StockLevel();
        pStockLevel = m_pTxn-
        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);
        >pvData;
        pData = (COM_DATA*)txn_out->parray-
        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();
    }
}

```

```

    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
    {
        >pvData;
        pData = (COM_DATA*)txn_in.parray-
        >BuffAddr_OrderStatus();
        pOrderStatus = m_pTxn-
        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);
        >pvData;
        pData = (COM_DATA*)txn_out->parray-
        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 <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 <atcom.h>
#include <initguid.h>
#include <transact.h>
//#include <atimpl.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
definitions 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"));

        _sntprintf(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. GetProcAddress 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)
        sprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

```

```

        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 */

#ifdef __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 "oidl.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 "oidl.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;
};

```

```

\tbcc_com_all\src\tbcc_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"

```

```

#ifdef APSTUDIO_READONLY_SYMBOLS
////////////////////////////////////
// English (U.S.) resources

#ifndef _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

#ifdef _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 //////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tib"

////////////////////////////////////
#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'
    }
    "%MODULE%"
    {
        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_HEADER( )

#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)

#ifdef !_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_HEADERING( )

#ifdef _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 "oidl.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;

#ifdef __cplusplus && !defined(CINTERFACE)

MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
public:
    virtual HRESULT STDMETHODCALLTYPE NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT STDMETHODCALLTYPE Payment(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT STDMETHODCALLTYPE Delivery(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT STDMETHODCALLTYPE StockLevel(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT STDMETHODCALLTYPE OrderStatus(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT STDMETHODCALLTYPE 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 ( STDMETHODCALLTYPE *NewOrder )(
        ITPCC * This,

        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *Payment )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *Delivery )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *StockLevel )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *OrderStatus )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( STDMETHODCALLTYPE *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,bxn_in,bxn_out) \
(This)->lpVtbl-> NewOrder(This,bxn_in,bxn_out)

#define ITPCC_Payment(This,bxn_in,bxn_out) \
(This)->lpVtbl-> Payment(This,bxn_in,bxn_out)

#define ITPCC_Delivery(This,bxn_in,bxn_out) \
(This)->lpVtbl-> Delivery(This,bxn_in,bxn_out)

#define ITPCC_StockLevel(This,bxn_in,bxn_out) \
(This)->lpVtbl-> StockLevel(This,bxn_in,bxn_out)

#define ITPCC_OrderStatus(This,bxn_in,bxn_out) \
(This)->lpVtbl-> OrderStatus(This,bxn_in,bxn_out)

#define ITPCC_CallSetComplete(This) \
(This)->lpVtbl-> CallSetComplete(This)

#endif /* COBJMACROS */

#endif /* C style interface */

```

```

HRESULT __stdcall ITPCC_NewOrder_Proxy(
ITPCC * This,
/* [in] */ VARIANT bxn_in,
/* [out] */ VARIANT *bxn_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 bxn_in,
/* [out] */ VARIANT *bxn_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 bxn_in,
/* [out] */ VARIANT *bxn_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 bxn_in,
/* [out] */ VARIANT *bxn_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 bxn_in,
/* [out] */ VARIANT *bxn_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
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_s1.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
    DllGetObject          @1 PRIVATE
    DllCanUnloadNow      @2          PRIVATE
    GetProxyDllInfo      @3 PRIVATE
    DllRegisterServer    @4
    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*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440

```

```

#endif
#include "rpc.h"
#include "rpcndr.h"

#ifdef __RPCNDR_H_VERSION__

#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifdef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#ifdef _MSC_VER && (_MSC_VER >= 1020)
#pragma once
#endif
#endif

/* Forward Declarations */

#ifdef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#ifdef __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;

#ifdef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

```

```

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

```

```

EXTERN_C const IID IID_ITPCC;

```

```

#ifdef __cplusplus && !defined(CINTERFACE)

```

```

MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
ITPCC : public IUnknown

```

```

{
public:
virtual HRESULT STDMETHODCALLTYPE NewOrder(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

```

```

virtual HRESULT STDMETHODCALLTYPE Payment(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

```

```

virtual HRESULT STDMETHODCALLTYPE Delivery(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

```

```

virtual HRESULT STDMETHODCALLTYPE StockLevel(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

```

```

virtual HRESULT STDMETHODCALLTYPE OrderStatus(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

```

```

virtual HRESULT STDMETHODCALLTYPE 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 ( STDMETHODCALLTYPE *NewOrder )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

    HRESULT ( STDMETHODCALLTYPE *Payment )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

    HRESULT ( STDMETHODCALLTYPE *Delivery )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

    HRESULT ( STDMETHODCALLTYPE *StockLevel )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

    HRESULT ( STDMETHODCALLTYPE *OrderStatus )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

    HRESULT ( STDMETHODCALLTYPE *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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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
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 "oidl.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,0xFEEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
#endif

```

```

#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_HEADERING( )

#ifndef _M_IA64 && !defined(_M_AMD64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this
file*/
#ifndef __REQD_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifdef __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 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 ];

#ifndef __RPC_WIN32__
#error Invalid build platform for this stub.
#endif

#ifdef 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 -Oicf or -Oicf, [wire_marshall] or [user_marshall] 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 */

        FC_AUTO_HANDLE /*/
        0x33,
        /*/
        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, */
/*
/* 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 */
/* 0, /* 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, */
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 */
NdrFcShort( 0x70 ), /* Flags: out, return, base
type, */
/* 188 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 190 */ 0x8, /* FC_LONG */
0x0, /* 0 */
*/

}
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
0,
{
*/
/* 2 */
/* 4 */ NdrFcShort( 0x3ca ), /* FC_UP */
/* 6 */ /* Offset= 970 (974) */

FC_NON_ENCAPSULATED_UNION */
0x2b, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /*
*/
/* 10 */ NdrFcShort( 0xffff ), /* -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 */ 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) */	/* 312 */	0x1b,	/*	/* 358 */	NdrFcShort( 0x0 ),	/* 0 */
/* 204 */	NdrFcLong( 0x10 ),	/* 16 */	FC_CARRAY */		/* 1	/* 360 */	0xc0,	/* 192 */
/* 208 */	NdrFcShort( 0x8002 ),	/* Simple arm type: FC_CHAR	*/	0x1,		*/		0x0,
*/			*/			*/		/* 0
/* 210 */	NdrFcLong( 0x12 ),	/* 18 */	/* 314 */	NdrFcShort( 0x2 ),	/* 2 */	/* 362 */	0x0,	/* 0 */
/* 214 */	NdrFcShort( 0x8006 ),	/* Simple arm type: FC_SHORT	/* 316 */	0x9,	/* Corr desc: FC_ULONG */	*/		0x0,
*/			*/			*/		/* 0
/* 216 */	NdrFcLong( 0x13 ),	/* 19 */	*/		0x0,	/*		
/* 220 */	NdrFcShort( 0x8008 ),	/* Simple arm type: FC_LONG	*/			/* 364 */	0x0,	/* 0 */
*/			/* 318 */	NdrFcShort( 0xfffc ),	/* -4 */	*/		0x0,
/* 222 */	NdrFcLong( 0x15 ),	/* 21 */	/* 320 */	0x6,	/* FC_SHORT */	/* 366 */	0x0,	/* 0 */
/* 226 */	NdrFcShort( 0x800b ),	/* Simple arm type: FC_HYPER			0x5b,	/*		0x46,
*/			FC_END */			*/		/*
/* 228 */	NdrFcLong( 0x16 ),	/* 22 */	/* 322 */		0x17,	70 */		
/* 232 */	NdrFcShort( 0x8008 ),	/* Simple arm type: FC_LONG	FC_CSTRUCT */			/* 368 */		
*/			*/		0x3,	/* 3		0x12, 0x10, /* FC_UP
/* 234 */	NdrFcLong( 0x17 ),	/* 23 */	*/			[pointer_deref] */		
/* 238 */	NdrFcShort( 0x8008 ),	/* Simple arm type: FC_LONG	/* 324 */	NdrFcShort( 0x8 ),	/* 8 */	/* 370 */	NdrFcShort( 0x2 ),	/* Offset= 2 (372) */
*/			/* 326 */	NdrFcShort( 0xfffff2 ),	/* Offset= -14 (312) */	/* 372 */		0x12, 0x0, /* FC_UP */
/* 240 */	NdrFcLong( 0xe ),	/* 14 */	/* 328 */	0x8,	/* FC_LONG */	/* 374 */	NdrFcShort( 0x1fc ),	/* Offset= 508 (882) */
/* 244 */	NdrFcShort( 0x2c8 ),	/* Offset= 712 (956) */			0x8,	/* 376 */		/*
/* 246 */	NdrFcLong( 0x400e ),	/* 16398 */	FC_LONG */			FC_ENCAPSULATED_UNION */		0x2a,
/* 250 */	NdrFcShort( 0x2cc ),	/* Offset= 716 (966) */	/* 330 */	0x5c,	/* FC_PAD */			/*
/* 252 */	NdrFcLong( 0x4010 ),	/* 16400 */	FC_END */		0x5b,	73 */		
/* 256 */	NdrFcShort( 0x2ca ),	/* Offset= 714 (970) */	/* 332 */			/* 378 */	NdrFcShort( 0x18 ),	/* 24 */
/* 258 */	NdrFcLong( 0x4012 ),	/* 16402 */	FC_IP */		0x2f,	/* 380 */	NdrFcShort( 0xa ),	/* 10 */
/* 262 */	NdrFcShort( 0x286 ),	/* Offset= 646 (908) */				/* 382 */	NdrFcLong( 0x8 ),	/* 8 */
/* 264 */	NdrFcLong( 0x4013 ),	/* 16403 */	FC_CONSTANT_IID */		0x5a,	/* 386 */	NdrFcShort( 0x58 ),	/* Offset= 88 (474) */
/* 268 */	NdrFcShort( 0x284 ),	/* Offset= 644 (912) */	/* 334 */	NdrFcLong( 0x0 ),	/* 0 */	/* 388 */	NdrFcLong( 0xd ),	/* 13 */
/* 270 */	NdrFcLong( 0x4015 ),	/* 16405 */	/* 336 */	NdrFcShort( 0x0 ),	/* 0 */	/* 392 */	NdrFcShort( 0x78 ),	/* Offset= 120 (512) */
/* 274 */	NdrFcShort( 0x282 ),	/* Offset= 642 (916) */	/* 338 */	NdrFcShort( 0x0 ),	/* 0 */	/* 394 */	NdrFcLong( 0x9 ),	/* 9 */
/* 276 */	NdrFcLong( 0x4016 ),	/* 16406 */	/* 340 */	NdrFcShort( 0x0 ),	/* 0 */	/* 398 */	NdrFcShort( 0x94 ),	/* Offset= 148 (546) */
/* 280 */	NdrFcShort( 0x278 ),	/* Offset= 632 (912) */	/* 342 */	0xc0,	/* 192 */	/* 400 */	NdrFcLong( 0xc ),	/* 12 */
/* 282 */	NdrFcLong( 0x4017 ),	/* 16407 */	*/		0x0,	/* 404 */	NdrFcShort( 0xbc ),	/* Offset= 188 (592) */
/* 286 */	NdrFcShort( 0x272 ),	/* Offset= 626 (912) */	/* 344 */	0x0,	/* 0 */	/* 406 */	NdrFcLong( 0x24 ),	/* 36 */
/* 288 */	NdrFcLong( 0x0 ),	/* 0 */	*/		0x0,	/* 410 */	NdrFcShort( 0x114 ),	/* Offset= 276 (686) */
/* 292 */	NdrFcShort( 0x0 ),	/* Offset= 0 (292) */	*/		0x0,	/* 412 */	NdrFcLong( 0x800d ),	/* 32781 */
/* 294 */	NdrFcLong( 0x1 ),	/* 1 */	/* 346 */	0x0,	/* 0 */	/* 416 */	NdrFcShort( 0x130 ),	/* Offset= 304 (720) */
/* 298 */	NdrFcShort( 0x0 ),	/* Offset= 0 (298) */	*/		0x0,	/* 418 */	NdrFcLong( 0x10 ),	/* 16 */
/* 300 */	NdrFcShort( 0xfffffff ),	/* Offset= -1 (299) */	/* 348 */	0x0,	/* 0 */	/* 422 */	NdrFcShort( 0x148 ),	/* Offset= 328 (750) */
/* 302 */			*/		0x46,	/* 424 */	NdrFcLong( 0x2 ),	/* 2 */
FC_STRUCT */		0x15, /*	*/			/* 428 */	NdrFcShort( 0x160 ),	/* Offset= 352 (780) */
*/			70 */			/* 430 */	NdrFcLong( 0x3 ),	/* 3 */
/* 304 */	NdrFcShort( 0x8 ),	/* 8 */	/* 350 */		0x2f,	/* 434 */	NdrFcShort( 0x178 ),	/* Offset= 376 (810) */
/* 306 */	0xb,	/* FC_HYPER */	FC_IP */			/* 436 */	NdrFcLong( 0x14 ),	/* 20 */
		0x5b, /*	FC_CONSTANT_IID */		0x5a,	/* 440 */	NdrFcShort( 0x190 ),	/* Offset= 400 (840) */
FC_END */			/* 352 */	NdrFcLong( 0x20400 ),	/* 132096 */	/* 442 */	NdrFcShort( 0xfffffff ),	/* Offset= -1 (441) */
/* 308 */		0x12, 0x0, /* FC_UP */	FC_CARRAY */			/* 444 */		0x1b,
*/			*/					/*
/* 310 */	NdrFcShort( 0xc ),	/* Offset= 12 (322) */						

```

0x3, /* 3
*/
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */

0x0, /*
*/
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */

FC_PP */
0x4b, /*
*/
FC_PAD */
0x5c, /*
*/
/* 454 */

FC_VARIABLE_REPEAT */
0x48, /*
*/

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( 0xffff6e ), /* Offset= -146 (322) */
/* 470 */

0x5b, /*
*/

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 */

```

```

/* 488 */ NdrFcShort( 0xffff44 ), /* Offset= -44 (444) */
/* 490 */

FC_END */
0x5b, /*
*/

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( 0xffff50 ), /* Offset= -176 (332) */
/* 510 */ 0x5c, /* FC_PAD */
0x5b, /*
*/

FC_END */
/* 512 */

FC_BOGUS_STRUCT */
0x1a, /*
*/

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( 0xffffe0 ), /* Offset= -32 (494) */
/* 528 */

0x21, /*
*/

FC_BOGUS_ARRAY */

```

```

0x3, /* 3
*/
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */

0x0, /*
*/

0x0, /*
*/
*/
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

0x0, /* 0
*/
*/
/* 542 */ NdrFcShort( 0xffff40 ), /* 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( 0xffffe0 ), /* 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 */

```

FC_VARIABLE_REPEAT */	0x48,	/*	*/ 624 */	0x0,	/* 0 */	/* 664 */	0x4b,	/*
	0x49,	/*	70 */	0x46,	/*	FC_PP */	0x5c,	/*
FC_FIXED_OFFSET */			/* 626 */			FC_PAD */		
/* 574 */	NdrFcShort( 0x4 ),	/* 4 */		0x1b,	/*	/* 666 */		
/* 576 */	NdrFcShort( 0x0 ),	/* 0 */					0x48,	/*
/* 578 */	NdrFcShort( 0x1 ),	/* 1 */	FC_CARRAY */					
/* 580 */	NdrFcShort( 0x0 ),	/* 0 */		0x0,	/* 0	FC_VARIABLE_REPEAT */		
/* 582 */	NdrFcShort( 0x0 ),	/* 0 */					0x49,	/*
/* 584 */	0x12, 0x0, /* FC_UP */		*/			FC_FIXED_OFFSET */		
/* 586 */	NdrFcShort( 0x184 ),	/* Offset= 388 (974) */	/* 628 */	NdrFcShort( 0x1 ),	/* 1 */	/* 668 */	NdrFcShort( 0x4 ),	/* 4 */
/* 588 */			/* 630 */	0x19,	/* Corr desc: field pointer,	/* 670 */	NdrFcShort( 0x0 ),	/* 0 */
FC_END */	0x5b,	/*	FC_ULONG */			/* 672 */	NdrFcShort( 0x1 ),	/* 1 */
				0x0,	/*	/* 674 */	NdrFcShort( 0x0 ),	/* 0 */
						/* 676 */	NdrFcShort( 0x0 ),	/* 0 */
FC_LONG */	0x8,	/*	*/			/* 678 */	0x12, 0x0, /* FC_UP */	
/* 590 */	0x5c,	/* FC_PAD */	/* 632 */	NdrFcShort( 0x4 ),	/* 4 */	/* 680 */	NdrFcShort( 0xfffffd4 ),	/* Offset= -44 (636) */
	0x5b,	/*	/* 634 */	0x1,	/* FC_BYTE */	/* 682 */		
FC_END */			FC_END */		/*			
/* 592 */			/* 636 */			FC_END */	0x5b,	/*
FC_BOGUS_STRUCT */	0x1a,	/*		0x1a,	/*			
	0x3,	/* 3	FC_BOGUS_STRUCT */				0x8,	/*
*/				0x3,	/* 3	FC_LONG */		
/* 594 */	NdrFcShort( 0x8 ),	/* 8 */	*/			/* 684 */	0x5c,	/* FC_PAD */
/* 596 */	NdrFcShort( 0x0 ),	/* 0 */	/* 638 */	NdrFcShort( 0x10 ),	/* 16 */			
/* 598 */	NdrFcShort( 0x6 ),	/* Offset= 6 (604) */	/* 640 */	NdrFcShort( 0x0 ),	/* 0 */	FC_END */		
/* 600 */	0x8,	/* FC_LONG */	/* 642 */	NdrFcShort( 0xa ),	/* Offset= 10 (652) */	/* 686 */		
	0x36,	/*	/* 644 */	0x8,	/* FC_LONG */		0x1a,	/*
FC_POINTER */			FC_LONG */		/*	FC_BOGUS_STRUCT */		
/* 602 */	0x5c,	/* FC_PAD */	/* 646 */	0x4c,	/* FC_EMBEDDED_COMPLEX	*/	0x3,	/* 3
	0x5b,	/*	*/			/* 688 */	NdrFcShort( 0x8 ),	/* 8 */
FC_END */				0x0,	/* 0	/* 690 */	NdrFcShort( 0x0 ),	/* 0 */
/* 604 */			*/			/* 692 */	NdrFcShort( 0x6 ),	/* Offset= 6 (698) */
/* 606 */	NdrFcShort( 0xfffffd4 ),	/* FC_RP */	/* 648 */	NdrFcShort( 0xfffffd8 ),	/* Offset= -40 (608) */	/* 694 */	0x8,	/* FC_LONG */
/* 608 */		/* Offset= -44 (562) */	/* 650 */	0x36,	/* FC_POINTER */			/*
FC_IP */	0x2f,	/*	FC_END */		/*	FC_POINTER */		
	0x5a,	/*	/* 652 */			/* 696 */	0x5c,	/* FC_PAD */
FC_CONSTANT_IID */			/* 654 */	NdrFcShort( 0xfffffe4 ),	/* Offset= -28 (626) */	FC_END */		
/* 610 */	NdrFcLong( 0x2f ),	/* 47 */	/* 656 */			/* 698 */		
/* 614 */	NdrFcShort( 0x0 ),	/* 0 */		0x12, 0x0, /* FC_UP */			0x11, 0x0, /* FC_RP */	
/* 616 */	NdrFcShort( 0x0 ),	/* 0 */	FC_CARRAY */		/*	/* 700 */	NdrFcShort( 0xfffffd4 ),	/* Offset= -44 (656) */
/* 618 */	0xc0,	/* 192 */		0x1b,	/*	/* 702 */		
*/	0x0,	/* 0	*/	0x3,	/* 3	FC_SMFARRAY */		
/* 620 */	0x0,	/* 0 */	/* 658 */	NdrFcShort( 0x4 ),	/* 4 */		0x1d,	/*
	0x0,	/* 0 */	/* 660 */	0x19,	/* Corr desc: field pointer,	*/	0x0,	/* 0
*/			FC_ULONG */			/* 704 */	NdrFcShort( 0x8 ),	/* 8 */
/* 622 */	0x0,	/* 0 */		0x0,	/*	/* 706 */	0x1,	/* FC_BYTE */
			/* 662 */	NdrFcShort( 0x0 ),	/* 0 */	FC_END */		
							0x5b,	/*

```

/* 708 */
FC_STRUCT */
0x15, /*
0x3, /* 3
*/
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT */
/* 714 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 716 */ 0x0, /* 0 */
NdrFcShort( 0xfffff1 ), /*
Offset= -15 (702) */
0x5b, /*
FC_END */
/* 720 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3
*/
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0xa ), /* Offset= 10 (736) */
/* 728 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 730 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
*/
0x0, /* 0
*/
/* 732 */ NdrFcShort( 0xffffe8 ), /* Offset= -24 (708) */
/* 734 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 736 */
0x11, 0x0, /* FC_RP */
/* 738 */ NdrFcShort( 0xffff0c ), /* Offset= -244 (494) */
/* 740 */
0x1b, /*
FC_CARRAY */
0x0, /* 0
*/
/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr desc: field pointer, */
FC_ULONG */
0x0, /*
*/
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /* FC_BYTE */

```

```

FC_END */
0x5b, /*
0x16, /*
FC_PSTRUCT */
0x3, /* 3
*/
/* 752 */ 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( 0xffffe8 ), /* 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, /*
0x5c, /*
FC_PP */
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( 0xffffe8 ), /* 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 */

```

FC_PAD */	0x5c,	/*	FC_LONG */	0x8,	/*	/* 896 */	NdrFcShort( 0xffffdf8 ),	0x0,	/* 0
/* 818 */	NdrFcShort( 0x4 ),	/* 4 */	/* 858 */	0x8,	/* FC_LONG */	/* 898 */	0x5c,	/* Offset= -520 (376) */	/* FC_PAD */
/* 820 */	NdrFcShort( 0x4 ),	/* 4 */			0x5b,			0x5b,	/*
/* 822 */	0x12, 0x0,	/* FC_UP */	FC_END */			FC_END */			
/* 824 */	NdrFcShort( 0xfffffe8 ),	/* Offset= -24 (800) */	/* 860 */		0x15,	/* 900 */		0x12, 0x0,	/* FC_UP */
/* 826 */			FC_STRUCT */			/* 902 */	NdrFcShort( 0xfffffef6 ),	/* Offset= -266 (636) */	/*
FC_END */	0x5b,	/*	*/	0x3,	/* 3	/* 904 */			
								0x12, 0x8,	/* FC_UP
FC_LONG */	0x8,	/*	/* 862 */	NdrFcShort( 0x8 ),	/* 8 */	[simple_pointer] */			
/* 828 */	0x8,	/* FC_LONG */	/* 864 */	0x8,	/* FC_LONG */	/* 906 */	0x1,	/* FC_BYTE */	/*
FC_END */		/*			0x8,			0x5c,	/*
/* 830 */			FC_LONG */			FC_PAD */			
	0x1b,	/*	/* 866 */	0x5c,	/* FC_PAD */	/* 908 */		0x12, 0x8,	/* FC_UP
FC_CARRAY */			FC_END */		0x5b,				
*/	0x7,	/* 7	/* 868 */			[simple_pointer] */			
/* 832 */	NdrFcShort( 0x8 ),	/* 8 */	FC_CARRAY */		0x1b,	/* 910 */	0x6,	/* FC_SHORT */	/*
/* 834 */	0x19,	/* Corr desc: field pointer,	*/			FC_PAD */		0x5c,	/*
FC_ULONG */			/* 870 */	NdrFcShort( 0x8 ),	/* 8 */	/* 912 */		0x12, 0x8,	/* FC_UP
*/	0x0,	/*	/* 872 */	0x7,	/* Corr desc: FC_USHORT */	[simple_pointer] */			
/* 836 */	NdrFcShort( 0x0 ),	/* 0 */	*/		0x0,	/* 914 */	0x8,	/* FC_LONG */	/*
/* 838 */	0xb,	/* FC_HYPER */	/* 874 */	NdrFcShort( 0xffd8 ),	/* -40 */	FC_PAD */		0x5c,	/*
FC_END */		/*	/* 876 */	0x4c,	/* FC_EMBEDDED_COMPLEX	/* 916 */		0x12, 0x8,	/* FC_UP
/* 840 */			*/		0x0,	[simple_pointer] */			
	0x16,	/*	/* 878 */	NdrFcShort( 0xfffffee ),	/* Offset= -18 (860) */	/* 918 */	0xb,	/* FC_HYPER */	/*
FC_PSTRUCT */			/* 880 */	0x5c,	/* FC_PAD */	/* 920 */		0x5c,	/*
*/	0x3,	/* 3	FC_END */		0x5b,			0x12, 0x8,	/* FC_UP
/* 842 */	NdrFcShort( 0x8 ),	/* 8 */	/* 882 */			[simple_pointer] */			
/* 844 */					0x1a,	/* 922 */	0xa,	/* FC_FLOAT */	/*
FC_PP */	0x4b,	/*	FC_BOGUS_STRUCT */			FC_PAD */		0x5c,	/*
	0x5c,	/*	*/		0x3,	/* 924 */			
FC_PAD */			/* 884 */	NdrFcShort( 0x28 ),	/* 40 */			0x12, 0x8,	/* FC_UP
FC_NO_REPEAT */	0x46,	/*	/* 886 */	NdrFcShort( 0xfffffee ),	/* Offset= -18 (868) */	[simple_pointer] */			
			/* 888 */	NdrFcShort( 0x0 ),	/* Offset= 0 (888) */	/* 926 */	0xc,	/* FC_DOUBLE */	/*
	0x5c,	/*	/* 890 */	0x6,	/* FC_SHORT */	FC_PAD */		0x5c,	/*
FC_PAD */					0x6,	/* 928 */			
/* 848 */	NdrFcShort( 0x4 ),	/* 4 */	FC_SHORT */					0x12, 0x0,	/* FC_UP */
/* 850 */	NdrFcShort( 0x4 ),	/* 4 */	/* 892 */	0x8,	/* FC_LONG */	/* 930 */	NdrFcShort( 0xfffffd8c ),	/* Offset= -628 (302) */	/*
/* 852 */	0x12, 0x0,	/* FC_UP */	FC_LONG */		0x8,	/* 932 */		0x12, 0x10,	/* FC_UP
/* 854 */	NdrFcShort( 0xfffffe8 ),	/* Offset= -24 (830) */	/* 894 */	0x4c,	/* FC_EMBEDDED_COMPLEX	[pointer_deref] */			
/* 856 */			*/			/* 934 */	NdrFcShort( 0xfffffd8e ),	/* Offset= -626 (308) */	/*
FC_END */	0x5b,	/*							

```

/* 936 */
[pointer_deref] */
/* 938 */ NdrFcShort( 0xffffda2 ), /* Offset= -606 (332) */
/* 940 */
[pointer_deref] */
/* 942 */ NdrFcShort( 0xffffdb0 ), /* Offset= -592 (350) */
/* 944 */
[pointer_deref] */
/* 946 */ NdrFcShort( 0xffffdbe ), /* Offset= -578 (368) */
/* 948 */
[pointer_deref] */
/* 950 */ NdrFcShort( 0x2 ), /* Offset= 2 (952) */
/* 952 */
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
FC_STRUCT */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /* FC_SHORT */
FC_BYTE */
/* 962 */ 0x1, /* FC_BYTE */
FC_LONG */
/* 964 */ 0xb, /* FC_HYPER */
FC_END */
/* 966 */
/* 968 */ NdrFcShort( 0xfffff4 ), /* Offset= -12 (956) */
/* 970 */
[simple_pointer] */
/* 972 */ 0x2, /* FC_CHAR */
FC_PAD */
/* 974 */
FC_BOGUS_STRUCT */
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ), /* 0 */
/* 980 */ NdrFcShort( 0x0 ), /* Offset= 0 (980) */
/* 982 */ 0x8, /* FC_LONG */

```

```

FC_SHORT */ 0x6, /* FC_SHORT */
FC_SHORT */ 0x6, /* FC_SHORT */
FC_SHORT */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/* 990 */ NdrFcShort( 0xffffc28 ), /* Offset= -984 (6) */
/* 992 */ 0x5c, /* FC_PAD */
FC_END */
/* 994 */ 0xb4, /* FC_USER_MARSHAL */
131 */
/* 996 */ NdrFcShort( 0x0 ), /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
/* 1002 */ NdrFcShort( 0xffffc18 ), /* Offset= -1000 (2) */
/* 1004 */
[allocated_on_stack] */
/* 1006 */ NdrFcShort( 0x6 ), /* Offset= 6 (1012) */
/* 1008 */
/* 1010 */ NdrFcShort( 0xffffdc ), /* Offset= -36 (974) */
/* 1012 */ 0xb4, /* FC_USER_MARSHAL */
131 */
/* 1014 */ NdrFcShort( 0x0 ), /* 0 */
/* 1016 */ NdrFcShort( 0x10 ), /* 16 */
/* 1018 */ NdrFcShort( 0x0 ), /* 0 */
/* 1020 */ NdrFcShort( 0xfffff4 ), /* Offset= -12 (1008) */
}
};
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
}

```

```

};
/* Standard interface: __MIDL_itf_tpcCom_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
,0x00,0x46}} */
/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{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
}

```



```

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,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    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[] =
{

```

```

    0 /* CInterfaceProxyVtbl */ &_ITPCCProxyVtbl,
};

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( )

#ifdef _M_IA64 || defined(_M_AMD64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this
file*/
#define __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifdef __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 ];

#ifdef __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 */
    }
}

```

```

/* 21 */ 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,
{
*/
/* 2 */
/* 4 */
/* 6 */
FC_NON_ENCAPSULATED_UNION */
FC_ULONG */
/* 8 */
0x7,
*/
/* 10 */
/* 12 */
/* 14 */
/* 16 */
/* 18 */
/* 20 */
/* 24 */
*/
/* 26 */
/* 30 */
*/
/* 32 */
/* 36 */
*/
/* 38 */
/* 42 */
*/
/* 44 */
/* 48 */
*/
/* 50 */
/* 54 */
FC_DOUBLE */
/* 56 */
/* 60 */
*/
/* 62 */
/* 66 */
*/
/* 68 */
/* 72 */
/* 74 */
/* 78 */
FC_DOUBLE */
/* 80 */
/* 84 */
/* 86 */
NdrFcShort( 0x0 ), /* 0
0x12, 0x0, /* FC_UP */
/* Offset= 950 (954) */
0x2b, /*
0x9, /*
/* Corr desc: FC_USHORT */
0x0, /*
/* -8 */
/* Corr flags: early, */
/* Offset= 2 (16) */
/* 16 */
/* 47 */
/* 20 */
/* Simple arm type: FC_HYPER
*/
/* 3 */
/* Simple arm type: FC_LONG
*/
/* 17 */
/* Simple arm type: FC_BYTE
*/
/* 2 */
/* Simple arm type: FC_SHORT
*/
/* 4 */
/* Simple arm type: FC_FLOAT
*/
/* 5 */
/* Simple arm type:
*/
/* 11 */
/* Simple arm type: FC_SHORT
*/
/* 10 */
/* Simple arm type: FC_LONG
*/
/* 6 */
/* Offset= 232 (304) */
/* 7 */
/* Simple arm type:
*/
/* 8 */
/* Offset= 226 (310) */
/* 13 */
NdrFcShort( 0x0 ), /* 0
0x12, 0x0, /* FC_UP */
/* Offset= 950 (954) */
0x2b, /*
0x9, /*
/* Corr desc: FC_USHORT */
0x0, /*
/* -8 */
/* Corr flags: early, */
/* Offset= 2 (16) */
/* 16 */
/* 47 */
/* 20 */
/* Simple arm type: FC_HYPER
*/
/* 3 */
/* Simple arm type: FC_LONG
*/
/* 17 */
/* Simple arm type: FC_BYTE
*/
/* 2 */
/* Simple arm type: FC_SHORT
*/
/* 4 */
/* Simple arm type: FC_FLOAT
*/
/* 5 */
/* Simple arm type:
*/
/* 11 */
/* Simple arm type: FC_SHORT
*/
/* 10 */
/* Simple arm type: FC_LONG
*/
/* 6 */
/* Offset= 232 (304) */
/* 7 */
/* Simple arm type:
*/
/* 8 */
/* Offset= 226 (310) */
/* 13 */

```

```

/* 90 */
/* 96 */
/* 98 */
/* 102 */
/* 104 */
/* 108 */
*/
/* 110 */
/* 114 */
/* 116 */
/* 120 */
/* 122 */
/* 126 */
/* 128 */
/* 132 */
/* 134 */
/* 138 */
/* 140 */
/* 144 */
/* 146 */
/* 150 */
/* 152 */
/* 156 */
/* 158 */
/* 162 */
/* 164 */
/* 168 */
/* 170 */
/* 174 */
/* 176 */
/* 180 */
/* 182 */
/* 186 */
/* 188 */
/* 192 */
/* 194 */
/* 198 */
/* 200 */
/* 204 */
/* 206 */
/* 210 */
*/
/* 212 */
/* 216 */
*/
/* 218 */
/* 222 */
*/
/* 224 */
/* 228 */
*/
/* 230 */
NdrFcShort( 0x0 ), /* 0
NdrFcShort( 0x102 ), /*
NdrFcLong( 0x2000 ), /*
NdrFcShort( 0x10e ), /*
NdrFcLong( 0x24 ), /*
NdrFcShort( 0x304 ), /*
NdrFcLong( 0x4024 ), /*
NdrFcShort( 0x2fe ), /*
NdrFcLong( 0x4011 ), /*
NdrFcShort( 0x2fc ), /*
NdrFcLong( 0x4002 ), /*
NdrFcShort( 0x2fa ), /*
NdrFcLong( 0x4003 ), /*
NdrFcShort( 0x2f8 ), /*
NdrFcLong( 0x4014 ), /*
NdrFcShort( 0x2f6 ), /*
NdrFcLong( 0x4004 ), /*
NdrFcShort( 0x2f4 ), /*
NdrFcLong( 0x4005 ), /*
NdrFcShort( 0x2f2 ), /*
NdrFcLong( 0x400b ), /*
NdrFcShort( 0x2dc ), /*
NdrFcLong( 0x400a ), /*
NdrFcShort( 0x2da ), /*
NdrFcLong( 0x4006 ), /*
NdrFcShort( 0x2e4 ), /*
NdrFcLong( 0x4007 ), /*
NdrFcShort( 0x2da ), /*
NdrFcLong( 0x4008 ), /*
NdrFcShort( 0x2dc ), /*
NdrFcLong( 0x400d ), /*
NdrFcShort( 0x2da ), /*
NdrFcLong( 0x4009 ), /*
NdrFcShort( 0x2d8 ), /*
NdrFcLong( 0x6000 ), /*
NdrFcShort( 0x2d6 ), /*
NdrFcLong( 0x400c ), /*
NdrFcShort( 0x2d4 ), /*
NdrFcLong( 0x10 ), /*
NdrFcShort( 0x8002 ), /*
NdrFcLong( 0x12 ), /*
NdrFcShort( 0x8006 ), /*
NdrFcLong( 0x13 ), /*
NdrFcShort( 0x8008 ), /*
NdrFcLong( 0x15 ), /*
NdrFcShort( 0x800b ), /*
NdrFcLong( 0x16 ), /*
NdrFcShort( 0x246 (336) */
/* Offset= 258 (354) */
/* 8192 */
/* Offset= 270 (372) */
/* 36 */
/* Offset= 772 (880) */
/* 16420 */
/* Offset= 766 (880) */
/* 16401 */
/* Offset= 764 (884) */
/* 16386 */
/* Offset= 762 (888) */
/* 16387 */
/* Offset= 760 (892) */
/* 16404 */
/* Offset= 758 (896) */
/* 16388 */
/* Offset= 756 (900) */
/* 16389 */
/* Offset= 754 (904) */
/* 16395 */
/* Offset= 732 (888) */
/* 16394 */
/* Offset= 730 (892) */
/* 16390 */
/* Offset= 740 (908) */
/* 16391 */
/* Offset= 730 (904) */
/* 16392 */
/* Offset= 732 (912) */
/* 16397 */
/* Offset= 730 (916) */
/* 16393 */
/* Offset= 728 (920) */
/* 24576 */
/* Offset= 726 (924) */
/* 16396 */
/* Offset= 724 (928) */
/* 16 */
/* Simple arm type: FC_CHAR
*/
/* 18 */
/* Simple arm type: FC_SHORT
*/
/* 19 */
/* Simple arm type: FC_LONG
*/
/* 21 */
/* Simple arm type: FC_HYPER
*/
/* 22 */

```

```

/* 234 */
/* 236 */
/* 240 */
*/
/* 242 */
/* 246 */
*/
/* 248 */
/* 252 */
/* 254 */
/* 258 */
/* 260 */
/* 264 */
/* 266 */
/* 270 */
/* 272 */
/* 276 */
/* 278 */
/* 282 */
/* 284 */
/* 288 */
/* 290 */
/* 294 */
/* 296 */
/* 300 */
/* 302 */
/* 304 */
*/
FC_STRUCT */
*/
/* 306 */
/* 308 */
*/
FC_END */
/* 310 */
*/
/* 312 */
/* 314 */
*/
FC_CARRAY */
*/
/* 316 */
/* 318 */
*/
*/
/* 320 */
/* 322 */
/* 324 */
*/
FC_END */
NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG
NdrFcLong( 0x17 ), /* 23 */
NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG
NdrFcLong( 0xe ), /* 14 */
NdrFcShort( 0x2b2 ), /* Offset= 690 (936) */
NdrFcLong( 0x400e ), /* 16398 */
NdrFcShort( 0x2b6 ), /* Offset= 694 (946) */
NdrFcLong( 0x4010 ), /* 16400 */
NdrFcShort( 0x2b4 ), /* Offset= 692 (950) */
NdrFcLong( 0x4012 ), /* 16402 */
NdrFcShort( 0x270 ), /* Offset= 624 (888) */
NdrFcLong( 0x4013 ), /* 16403 */
NdrFcShort( 0x26e ), /* Offset= 622 (892) */
NdrFcLong( 0x4015 ), /* 16405 */
NdrFcShort( 0x26c ), /* Offset= 620 (896) */
NdrFcLong( 0x4016 ), /* 16406 */
NdrFcShort( 0x262 ), /* Offset= 610 (892) */
NdrFcLong( 0x4017 ), /* 16407 */
NdrFcShort( 0x25c ), /* Offset= 604 (892) */
NdrFcLong( 0x0 ), /* 0 */
NdrFcShort( 0x0 ), /* Offset= 0 (294) */
NdrFcLong( 0x1 ), /* 1 */
NdrFcShort( 0x0 ), /* Offset= 0 (300) */
NdrFcShort( 0xffffffff ), /* Offset= -1 (301) */
0x15, /*
0x7, /* 7
NdrFcShort( 0x8 ), /* 8 */
0xb, /* FC_HYPER */
0x5b, /*
NdrFcShort( 0xe ), /* FC_UP */
/* Offset= 14 (326) */
0x1b, /*
0x1, /* 1
NdrFcShort( 0x2 ), /* 2 */
0x9, /* Corr desc: FC_ULONG */
0x0, /*
NdrFcShort( 0xfffc ), /* -4 */
NdrFcShort( 0x1 ), /* Corr flags: early, */
0x6, /* FC_SHORT */
0x5b, /*

```

/* 326 */			0x46,	/*	/* 466 */ NdrFcShort( 0xffff74 ),	/* Offset= -140 (820) */	
FC_CSTRUCT */	0x17,	/*	0x12, 0x10, /* FC_UP		/* 468 */ 0x5c,	/* FC_PAD */	/*
*/	0x3,	/* 3	[pointer_deref] */		FC_END */		/*
/* 328 */ NdrFcShort( 0x8 ),	/* 8 */		/* 374 */ NdrFcShort( 0x2 ),	/* Offset= 2 (376) */	/* 470 */		
/* 330 */ NdrFcShort( 0xfffff0 ),	/* Offset= -16 (314) */		/* 376 */		FC_BOGUS_STRUCT */		/*
/* 332 */ 0x8,	/* FC_LONG */	/*		0x12, 0x0, /* FC_UP */			/*
FC_LONG */	0x8,	/*	/* 378 */ NdrFcShort( 0x1e4 ),	/* Offset= 484 (862) */			/*
/* 334 */ 0x5c,	/* FC_PAD */	/*	/* 380 */		/*		/* 3
FC_END */	0x5b,	/*	FC_ENCAPSULATED_UNION */		/* 472 */ NdrFcShort( 0x10 ),	/* 16 */	
/* 336 */				0x2a,	/* 474 */ NdrFcShort( 0x0 ),	/* 0 */	
FC_IP */	0x2f,	/*	137 */	0x89,	/* 476 */ NdrFcShort( 0x6 ),	/* Offset= 6 (482) */	
FC_CONSTANT_IID */	0x5a,	/*	/* 382 */ NdrFcShort( 0x20 ),	/* 32 */	/* 478 */ 0x8,	/* FC_LONG */	/*
/* 338 */ NdrFcLong( 0x0 ),	/* 0 */		/* 384 */ NdrFcShort( 0xa ),	/* 10 */	FC_STRUCTUREPAD4 */		/*
/* 342 */ NdrFcShort( 0x0 ),	/* 0 */		/* 386 */ NdrFcLong( 0x8 ),	/* 8 */	/* 480 */ 0x36,	/* FC_POINTER */	/*
/* 344 */ NdrFcShort( 0x0 ),	/* 0 */		/* 390 */ NdrFcShort( 0x50 ),	/* Offset= 80 (470) */	FC_END */		/*
/* 346 */ 0xc0,	/* 192 */	/* 0	/* 392 */ NdrFcLong( 0xd ),	/* 13 */	/* 482 */		
*/	0x0,	/* 0	/* 396 */ NdrFcShort( 0x70 ),	/* Offset= 112 (508) */	/* 484 */ NdrFcShort( 0xffffdc ),	/* Offset= -36 (448) */	
/* 348 */ 0x0,	/* 0 */	/* 0	/* 402 */ NdrFcShort( 0x90 ),	/* Offset= 144 (546) */	/* 486 */		/*
*/	0x0,	/* 0	/* 404 */ NdrFcShort( 0xc ),	/* 12 */	FC_BOGUS_ARRAY */		/* 3
/* 350 */ 0x0,	/* 0 */	/* 0	/* 408 */ NdrFcShort( 0xb0 ),	/* Offset= 176 (584) */	/*		
*/	0x0,	/* 0	/* 410 */ NdrFcLong( 0x24 ),	/* 36 */	/* 488 */ NdrFcShort( 0x0 ),	/* 0 */	
/* 352 */ 0x0,	/* 0 */	/* 0	/* 414 */ NdrFcShort( 0x102 ),	/* Offset= 258 (672) */	/* 490 */ 0x19,	/* Corr desc: field pointer,	
70 */			/* 416 */ NdrFcLong( 0x800d ),	/* 32781 */	FC_ULONG */		/*
/* 354 */			/* 420 */ NdrFcShort( 0x11e ),	/* Offset= 286 (706) */	/*		
FC_IP */	0x2f,	/*	/* 422 */ NdrFcLong( 0x10 ),	/* 16 */	/* 492 */ NdrFcShort( 0x0 ),	/* 0 */	
FC_CONSTANT_IID */	0x5a,	/*	/* 426 */ NdrFcShort( 0x138 ),	/* Offset= 312 (738) */	/* 494 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */	
/* 356 */ NdrFcLong( 0x20400 ),	/* 132096 */		/* 428 */ NdrFcLong( 0x2 ),	/* 2 */	/* 496 */ NdrFcLong( 0xfffff ),	/* -1 */	
/* 360 */ NdrFcShort( 0x0 ),	/* 0 */		/* 432 */ NdrFcShort( 0x14e ),	/* Offset= 334 (766) */	/* 500 */ NdrFcShort( 0x0 ),	/* Corr flags: */	
/* 362 */ NdrFcShort( 0x0 ),	/* 0 */		/* 434 */ NdrFcLong( 0x3 ),	/* 3 */	/* 502 */ 0x4c,	/* FC_EMBEDDED_COMPLEX	
/* 364 */ 0xc0,	/* 192 */	/* 0	/* 438 */ NdrFcShort( 0x164 ),	/* Offset= 356 (794) */	*/		/* 0
*/	0x0,	/* 0	/* 440 */ NdrFcLong( 0x14 ),	/* 20 */	/* 504 */ NdrFcShort( 0xfffff58 ),	/* Offset= -168 (336) */	
/* 366 */ 0x0,	/* 0 */	/* 0	/* 444 */ NdrFcShort( 0x17a ),	/* Offset= 378 (822) */	/* 506 */ 0x5c,	/* FC_PAD */	/*
*/	0x0,	/* 0	/* 446 */ NdrFcShort( 0xfffff ),	/* Offset= -1 (445) */	FC_END */		/*
/* 368 */ 0x0,	/* 0 */	/* 0	/* 448 */		/* 508 */		/*
*/	0x0,	/* 0	FC_BOGUS_ARRAY */		FC_BOGUS_STRUCT */		/* 3
/* 370 */ 0x0,	/* 0 */	/* 0	*/	0x21,	/*		
			/* 450 */ NdrFcShort( 0x0 ),	/* 0 */	/* 510 */ NdrFcShort( 0x10 ),	/* 16 */	
			/* 452 */ 0x19,	/* Corr desc: field pointer,	/* 512 */ NdrFcShort( 0x0 ),	/* 0 */	
			FC_ULONG */		/* 514 */ NdrFcShort( 0x6 ),	/* Offset= 6 (520) */	
			*/	0x0,			
			/* 454 */ NdrFcShort( 0x0 ),	/* 0 */			
			/* 456 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */			
			/* 458 */ NdrFcLong( 0xfffff ),	/* -1 */			
			/* 462 */ NdrFcShort( 0x0 ),	/* Corr flags: */			
			/* 464 */				

/* 516 */ 0x8,	/* FC_LONG */	0x3,	/* 3	0x46,	/*
FC_STRUCTPAD4 */	0x40,	/* 0 */		0x1b,	/*
/* 518 */ 0x36,	/* FC_POINTER */	/* Corr desc: field pointer,		FC_CARRAY */	
FC_END */	0x5b,	/*		0x0,	/* 0
/* 520 */		0x0,	/*	*/	
/* 522 */ NdrFcShort( 0xffffdc ),	0x11, 0x0, /* FC_RP */	/* 0 */		/* 620 */ NdrFcShort( 0x1 ),	/* 1 */
/* 524 */	/* Offset= -36 (486) */	/* 570 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */	/* 622 */ 0x19,	/* Corr desc: field pointer,
FC_BOGUS_ARRAY */	0x21,	/* 572 */ NdrFcLong( 0xffffffff ),	/* -1 */	FC_ULONG */	/*
*/	0x3,	/* 576 */ NdrFcShort( 0x0 ),	/* Corr flags: */	*/	0x0,
/* 526 */ NdrFcShort( 0x0 ),	/* 0 */	/* 578 */		/* 624 */ NdrFcShort( 0x4 ),	/* 4 */
/* 528 */ 0x19,	/* Corr desc: field pointer,	/* 580 */ NdrFcShort( 0x176 ),	/* Offset= 374 (954) */	/* 626 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */
FC_ULONG */	0x0,	/* 582 */ 0x5c,	/* FC_PAD */	/* 628 */ 0x1,	/* FC_BYTE */
*/	/*	FC_END */	0x5b,	FC_END */	0x5b,
/* 530 */ NdrFcShort( 0x0 ),	/* 0 */	/* 584 */	/*	/* 630 */	/*
/* 532 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */	FC_BOGUS_STRUCT */	0x1a,	FC_BOGUS_STRUCT */	0x1a,
/* 534 */ NdrFcLong( 0xffffffff ),	/* -1 */	*/	/*	*/	/*
/* 538 */ NdrFcShort( 0x0 ),	/* Corr flags: */	/* 586 */ NdrFcShort( 0x10 ),	0x3,	/* 632 */ NdrFcShort( 0x18 ),	/* 24 */
/* 540 */ 0x4c,	/* FC_EMBEDDED_COMPLEX	/* 588 */ NdrFcShort( 0x0 ),	/* 3	/* 634 */ NdrFcShort( 0x0 ),	/* 0 */
*/	0x0,	/* 590 */ NdrFcShort( 0x6 ),	/*	/* 636 */ NdrFcShort( 0xa ),	/* Offset= 10 (646) */
/* 542 */ NdrFcShort( 0xfffff44 ),	/* Offset= -188 (354) */	/* 592 */ 0x8,	/*	/* 638 */ 0x8,	/* FC_LONG */
/* 544 */ 0x5c,	/* FC_PAD */	FC_STRUCTPAD4 */	0x40,	FC_LONG */	0x8,
FC_END */	0x5b,	/* 594 */ 0x36,	/* FC_POINTER */	/* 640 */ 0x4c,	/* FC_EMBEDDED_COMPLEX
/* 546 */	/*	FC_END */	0x5b,	*/	0x0,
FC_BOGUS_STRUCT */	0x1a,	/* 596 */	/*	/* 642 */ NdrFcShort( 0xfffffd6 ),	/* Offset= -42 (600) */
*/	0x3,	/* 598 */ NdrFcShort( 0xffffdc ),	0x11, 0x0, /* FC_RP */	/* 644 */ 0x36,	/* FC_POINTER */
/* 548 */ NdrFcShort( 0x10 ),	/* 16 */	/* 600 */	/*	/* 646 */	0x5b,
/* 550 */ NdrFcShort( 0x0 ),	/* 0 */	FC_IP */	0x2f,	FC_END */	/*
/* 552 */ NdrFcShort( 0x6 ),	/* Offset= 6 (558) */	*/	/*	/* 648 */ NdrFcShort( 0xfffffe2 ),	0x12, 0x0, /* FC_UP */
/* 554 */ 0x8,	/* FC_LONG */	FC_CONSTANT_IID */	0x5a,	/* 650 */	/* Offset= -30 (618) */
FC_STRUCTPAD4 */	0x40,	/* 602 */ NdrFcLong( 0x2f ),	/* 47 */	FC_BOGUS_ARRAY */	0x21,
/* 556 */ 0x36,	/* FC_POINTER */	/* 606 */ NdrFcShort( 0x0 ),	/* 0 */	*/	/*
FC_END */	0x5b,	/* 608 */ NdrFcShort( 0x0 ),	/* 0 */	/* 652 */ NdrFcShort( 0x0 ),	/* 0 */
/* 558 */	/*	/* 610 */ 0xc0,	/* 192 */	/* 654 */ 0x19,	/* Corr desc: field pointer,
/* 560 */ NdrFcShort( 0xffffdc ),	0x11, 0x0, /* FC_RP */	*/	0x0,	FC_ULONG */	0x0,
/* 562 */	/* Offset= -36 (524) */	/* 612 */ 0x0,	/* 0 */	*/	/*
FC_BOGUS_ARRAY */	0x21,	/* 614 */ 0x0,	0x0,	/* 656 */ NdrFcShort( 0x0 ),	/* 0 */
*/	/*	*/	/*	/* 658 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */
/* 616 */ 0x0,	/*	/* 616 */ 0x0,	/* 0 */	/* 660 */ NdrFcLong( 0xffffffff ),	/* -1 */

```

/* 664 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 666 */
/* 668 */ NdrFcShort( 0xfffffda ), /* Offset= -38 (630) */
/* 670 */ 0x5c, /* FC_PAD */
/* 672 */ FC_END */
/* 674 */ FC_BOGUS_STRUCT */
/* 676 */ NdrFcShort( 0x10 ), /* 16 */
/* 678 */ NdrFcShort( 0x0 ), /* 0 */
/* 680 */ NdrFcShort( 0x6 ), /* Offset= 6 (684) */
/* 682 */ 0x8, /* FC_LONG */
/* 684 */ FC_STRUCTPAD4 */
/* 686 */ 0x36, /* FC_POINTER */
/* 688 */ FC_END */
/* 690 */ NdrFcShort( 0xfffffdc ), /* FC_RP */
/* 692 */ 0x1, /* Offset= -36 (650) */
/* 694 */ FC_SMFARRAY */
/* 696 */ NdrFcShort( 0x8 ), /* 8 */
/* 698 */ 0x1, /* FC_BYTE */
/* 700 */ FC_END */
/* 702 */ FC_STRUCT */
/* 704 */ NdrFcShort( 0x10 ), /* 16 */
/* 706 */ 0x8, /* FC_LONG */
/* 708 */ FC_SHORT */
/* 710 */ 0x6, /* FC_SHORT */
/* 712 */ FC_EMBEDDED_COMPLEX */
/* 714 */ 0x0, /* 0 */
/* 716 */ Offset= -15 (688) */
/* 718 */ FC_END */
/* 720 */ 0x5b, /* 5b */
/* 722 */ FC_END */
/* 724 */ 0x1a, /* 1a */
/* 726 */ 0x3, /* 3 */
/* 728 */ NdrFcShort( 0x10 ), /* 16 */
/* 730 */ 0x0, /* FC_SHORT */
/* 732 */ FC_EMBEDDED_COMPLEX */
/* 734 */ 0x0, /* 0 */
/* 736 */ Offset= -25 (694) */
/* 738 */ FC_END */
/* 740 */ 0x5b, /* 5b */
/* 742 */ FC_END */
/* 744 */ 0x11, 0x0, /* FC_RP */
/* 746 */ NdrFcShort( 0xfffff12 ), /* Offset= -238 (486) */
/* 748 */ FC_CARRAY */
/* 750 */ FC_ULONG */
/* 752 */ NdrFcShort( 0x1 ), /* 1 */
/* 754 */ 0x19, /* FC_ULONG */
/* 756 */ FC_END */
/* 758 */ NdrFcShort( 0x0 ), /* 0 */
/* 760 */ NdrFcShort( 0x1 ), /* 1 */
/* 762 */ 0x1, /* FC_BYTE */
/* 764 */ FC_END */
/* 766 */ FC_STRUCT */
/* 768 */ NdrFcShort( 0x10 ), /* 16 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x6 ), /* Offset= 6 (778) */
/* 774 */ 0x8, /* FC_LONG */
/* 776 */ FC_STRUCTPAD4 */
/* 778 */ 0x36, /* FC_POINTER */
/* 780 */ FC_END */
/* 782 */ NdrFcShort( 0xfffffe6 ), /* FC_ULONG */
/* 784 */ NdrFcShort( 0x4 ), /* 4 */
/* 786 */ 0x19, /* FC_ULONG */
/* 788 */ FC_ULONG */
/* 790 */ NdrFcShort( 0x0 ), /* 0 */
/* 792 */ NdrFcShort( 0x1 ), /* 1 */
/* 794 */ 0x8, /* FC_ULONG */
/* 796 */ FC_END */
/* 798 */ NdrFcShort( 0x1a ), /* 1a */
/* 800 */ FC_BOGUS_STRUCT */

```

```

FC_BOGUS_STRUCT */
0x1a, /* 1a */
0x3, /* 3 */
/* 708 */ NdrFcShort( 0x20 ), /* 32 */
/* 710 */ NdrFcShort( 0x0 ), /* 0 */
/* 712 */ NdrFcShort( 0xa ), /* Offset= 10 (722) */
/* 714 */ 0x8, /* FC_LONG */
/* 716 */ FC_STRUCTPAD4 */
/* 718 */ 0x36, /* FC_POINTER */
/* 720 */ FC_EMBEDDED_COMPLEX */
/* 722 */ 0x0, /* 0 */
/* 724 */ Offset= -25 (694) */
/* 726 */ FC_END */
/* 728 */ NdrFcShort( 0xfffff12 ), /* FC_RP */
/* 730 */ NdrFcShort( 0x1 ), /* 1 */
/* 732 */ NdrFcShort( 0x0 ), /* 0 */
/* 734 */ NdrFcShort( 0x1 ), /* 1 */
/* 736 */ 0x1, /* FC_BYTE */
/* 738 */ FC_END */
/* 740 */ FC_BOGUS_STRUCT */
/* 742 */ NdrFcShort( 0x10 ), /* 16 */
/* 744 */ NdrFcShort( 0x0 ), /* 0 */
/* 746 */ NdrFcShort( 0x6 ), /* Offset= 6 (750) */
/* 748 */ 0x8, /* FC_LONG */
/* 750 */ FC_STRUCTPAD4 */
/* 752 */ 0x36, /* FC_POINTER */
/* 754 */ FC_END */
/* 756 */ 0x12, 0x0, /* FC_UP */

```

```

/* 752 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (726) */
/* 754 */ FC_CARRAY */
/* 756 */ NdrFcShort( 0x2 ), /* 2 */
/* 758 */ 0x19, /* Corr desc: field pointer, */
/* 760 */ FC_ULONG */
/* 762 */ NdrFcShort( 0x0 ), /* 0 */
/* 764 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 766 */ 0x6, /* FC_SHORT */
/* 768 */ FC_END */
/* 770 */ NdrFcShort( 0x10 ), /* 16 */
/* 772 */ NdrFcShort( 0x0 ), /* 0 */
/* 774 */ NdrFcShort( 0x6 ), /* Offset= 6 (778) */
/* 776 */ 0x8, /* FC_LONG */
/* 778 */ FC_STRUCTPAD4 */
/* 780 */ 0x36, /* FC_POINTER */
/* 782 */ FC_END */
/* 784 */ NdrFcShort( 0xfffffe6 ), /* FC_ULONG */
/* 786 */ NdrFcShort( 0x0 ), /* 0 */
/* 788 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 790 */ 0x8, /* FC_LONG */
/* 792 */ FC_END */
/* 794 */ NdrFcShort( 0x1a ), /* 1a */
/* 796 */ FC_BOGUS_STRUCT */

```

	0x3,	/* 3	/* 842 */ 0x8,	<del>0x8</del> FC_LONG */	/*		0x12, 0x8, /* FC_UP
/*			FC_LONG */			[simple_pointer] */	/* FC_BYTE */
/* 796 */ NdrFcShort( 0x10 ),	/* 16 */		/* 844 */ 0x5c,	/* FC_PAD */	/*	FC_PAD */	/* 888 */
/* 798 */ NdrFcShort( 0x0 ),	/* 0 */			0x5b,	/*		0x5c, /*
/* 800 */ NdrFcShort( 0x6 ),	/* Offset= 6 (806) */		FC_END */			/* 888 */	
/* 802 */ 0x8,	/* FC_LONG */	/*	/* 846 */				0x12, 0x8, /* FC_UP
FC_STRUCTPAD4 */	0x40,	/*		0x1b,	/*	[simple_pointer] */	
/* 804 */ 0x36,	/* FC_POINTER */		FC_CARRAY */			/* 890 */ 0x6,	/* FC_SHORT */
	0x5b,	/*		0x3,	/* 3		0x5c, /*
FC_END */			/*			FC_PAD */	/* 892 */
/* 806 */			/* 848 */ NdrFcShort( 0x8 ),	/* 8 */		/* 892 */	
/* 808 */ NdrFcShort( 0xfffffe6 ),	0x12, 0x0, /* FC_UP */		/* 850 */ 0x7,	/* Corr desc: FC_USHORT */	/*	[simple_pointer] */	0x12, 0x8, /* FC_UP
/* 810 */	/* Offset= -26 (782) */			0x0,	/*	/* 894 */ 0x8,	/* FC_LONG */
			/*			/* 896 */	/*
FC_CARRAY */	0x1b,	/*	/* 852 */ NdrFcShort( 0xffc8 ),	/* -56 */		FC_PAD */	0x5c, /*
	0x7,	/* 7	/* 854 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */		/* 896 */	
/*			/* 856 */ 0x4c,	/* FC_EMBEDDED_COMPLEX			0x12, 0x8, /* FC_UP
/* 812 */ NdrFcShort( 0x8 ),	/* 8 */		/*			[simple_pointer] */	/* FC_UP
/* 814 */ 0x19,	/* Corr desc: field pointer,		/*	0x0,	/* 0	/* 898 */ 0xb,	/* FC_HYPER */
FC_ULONG */	0x0,	/*	/* 858 */ NdrFcShort( 0xfffffec ),	/* Offset= -20 (838) */		FC_PAD */	/* 900 */
/*			/* 860 */ 0x5c,	/* FC_PAD */	/*	/* 900 */	0x12, 0x8, /* FC_UP
/* 816 */ NdrFcShort( 0x0 ),	/* 0 */		FC_END */			[simple_pointer] */	
/* 818 */ NdrFcShort( 0x1 ),	/* Corr flags: early, */		/* 862 */			/* 902 */ 0xa,	/* FC_FLOAT */
/* 820 */ 0xb,	/* FC_HYPER */	/*		0x1a,	/*	/* 904 */	/*
FC_END */	0x5b,	/*	FC_BOGUS_STRUCT */		/* 3		0x5c, /*
/* 822 */			/*	0x3,		FC_PAD */	/* 904 */
FC_BOGUS_STRUCT */	0x1a,	/*	/* 864 */ NdrFcShort( 0x38 ),	/* 56 */			0x12, 0x8, /* FC_UP
	0x3,	/* 3	/* 866 */ NdrFcShort( 0xfffffec ),	/* Offset= -20 (846) */		[simple_pointer] */	
/*			/* 868 */ NdrFcShort( 0x0 ),	/* Offset= 0 (868) */		/* 906 */ 0xc,	/* FC_DOUBLE */
/* 824 */ NdrFcShort( 0x10 ),	/* 16 */		/* 870 */ 0x6,	/* FC_SHORT */	/*	FC_PAD */	/* 908 */
/* 826 */ NdrFcShort( 0x0 ),	/* 0 */			0x6,	/*	/* 908 */	
/* 828 */ NdrFcShort( 0x6 ),	/* Offset= 6 (834) */		FC_SHORT */				0x12, 0x0, /* FC_UP */
/* 830 */ 0x8,	/* FC_LONG */	/*	/* 872 */ 0x8,	/* FC_LONG */	/*	/* 910 */ NdrFcShort( 0xffffda2 ),	/* Offset= -606 (304) */
FC_STRUCTPAD4 */	0x40,	/*	FC_LONG */			/* 912 */	
/* 832 */ 0x36,	/* FC_POINTER */		/* 874 */ 0x40,	/* FC_STRUCTPAD4 */	/*		0x12, 0x10, /* FC_UP
	0x5b,	/*	FC_EMBEDDED_COMPLEX */	0x4c,	/*	[pointer_deref] */	
FC_END */			/* 876 */ 0x0,	/* 0 */		/* 914 */ NdrFcShort( 0xffffd4 ),	/* Offset= -604 (310) */
/* 834 */				NdrFcShort( 0xffffe0f ),	/*	/* 916 */	0x12, 0x10, /* FC_UP
/* 836 */ NdrFcShort( 0xfffffe6 ),	0x12, 0x0, /* FC_UP */		Offset= -497 (380) */			[pointer_deref] */	
/* 838 */	/* Offset= -26 (810) */			0x5b,	/*	/* 918 */ NdrFcShort( 0xffffdba ),	/* Offset= -582 (336) */
FC_STRUCT */	0x15,	/*	FC_END */			/* 920 */	0x12, 0x10, /* FC_UP
/*			/* 880 */			[pointer_deref] */	
/* 840 */ NdrFcShort( 0x8 ),	/* 8 */	/* 3	/* 882 */ NdrFcShort( 0xfffff04 ),	/* Offset= -252 (630) */		/* 922 */ NdrFcShort( 0xffffdc8 ),	/* Offset= -568 (354) */
			/* 884 */			/* 924 */	



```

[pointer_deref] */
/* 926 */ NdrFcShort( 0xffffdd6 ), /* 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( 0xfffff4 ), /* 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( 0xffffc3c ), /* 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( 0xffffc2c ), /* Offset= -980 (2) */
/* 984 */
0x11, 0x4, /* FC_RP
[allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /* Offset= 6 (992) */
/* 988 */
0x13, 0x0, /* FC_OP */
/* 990 */ NdrFcShort( 0xffffdc ), /* Offset= -36 (954) */
/* 992 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /*
131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xfffff4 ), /* Offset= -12 (988) */
}
};
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
{
VARIANT_UserSize
,VARIANT_UserMarshal
,VARIANT_UserUnmarshal
,VARIANT_UserFree
}
};
*/
Standard interface: __MIDL_itf_tpc_com_ps_0000, ver. 0.0,

```

```

GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

```

```

/* Object interface: IUnknown, ver. 0.0,

```

```

GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

```

```

/* Object interface: ITPCC, ver. 0.0,

```

```

GUID={0xFEE6AA2,0x84B1,0x11d2,{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::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,
    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[] =
{

```

```

    0(CInterfaceStubVtbl *) &_ITPCCStubVtbl,
};

```

```

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)*/

```

## Appendix B Database Load

### Setup.cmd

```
@echo off

set DoDBCREATE=TRUE
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=12
set WAREHOUSES=98000
set DB=tpcc
set PASSWD=ssdl

set HOMEPATH=%HOMEDRIVE%%HOMEDIR%
set FILEPATH=%FILEDRIVE%%FILEDIR%

if %PROCESSOR_ARCHITECTURE% == x86 (set
EXECUTABLEPATH=%HOMEPATH%\Executables\x86
) else set EXECUTABLEPATH=%HOMEPATH%\Executables\jpf

%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 -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 -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 -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 -
o%OUTPUTPATH%\NewCreateItemIndexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
echo Starting bulk inserts for Item
%EXECUTABLEPATH%\stockldr\tpccldr -s1 -W10 -titem -Pssdl -i0 -
f%OUTPUTPATH%\loaditem.out -L%OUTPUTPATH%\
if ERRORLEVEL 1 goto :ERROR_EXIT

echo Starting bulk inserts of item customer warehouse 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 %i
for /l %j in (1,1,%THREADS%) do start cmd /C
%HOMEPATH%\SetupOldLoader BULKINSERTn %1 %i %j
%EXECUTABLEPATH%\Semaphore -wait %THREADS%

echo Starting create indexes for %i
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreate%IIndexes.sql -o
%OUTPUTPATH%\NewCreate%IIndexes.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 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% -i0 -
f%OUTPUTPATH%\load%2%STARTWAREHOUSE%.out -L%OUTPUTPATH%\
%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 -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 -o %OUTPUTPATH%\%%i.out -b

osql -E -Q"sp_dboption %DB%,trunc,FALSE" -b
>%OUTPUTPATH%\dboptionTruncOFF.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 -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:          TPCC.H
 *
 *      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 id
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;
//synchronization 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;

        //synchronization 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_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;};
char *ErrorTypeStr() { return "WEBCLIENT"; }
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 DIIMain(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);

```

### tpccldr.c

```

// File: TPCCLDR.C Microsoft TPC-C Kit
// Ver. 4.42 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 DISTRICT_PER_WAREHOUSE 10
#define ORDERS_PER_DISTRICT 3000
#define ORDERS_SCALE_DOWN 30
#define MAX_CUSTOMER_THREADS 2
#define MAX_ORDER_THREADS 3
#define MAX_MAIN_THREADS 4

// Functions declarations
void HandleErrorDBC (SQLHDBC hdbc1);
void CheckDataBase();
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_i_id;
    ol;
}

```

```

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;
    // fix to avoid ODBC float to numeric conversion problem.
    // double c_balance;
    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];
long
c_id;
} CUSTOMER_SORT_STRUCT;

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
totaldberrors = 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
max_items;
long
customers_per_district;
long
orders_per_district;
long
first_new_order;
long
last_new_order;

TPCCCLDR_ARGS *aptr, args;

//=====
//
// Function name: main
//
//=====

int main(int argc, char **argv)
{
DWORD
dwThreadID[MAX_MAIN_THREADS];
HANDLE
hThread[MAX_MAIN_THREADS];
FILE
*floder;
char
buffer[255];
int
i;

for (i=0; i<MAX_MAIN_THREADS; i++)
hThread[i] = NULL;

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");
}

// 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 = 300;
}
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);

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()
{
    long          long          i_id;
    long          long          i_im_id;
    char          char          i_name[I_NAME_LEN+1];
    double        double        i_price;
    char          char          i_data[I_DATA_LEN+1];
    char          char          name[20];
    long          long          time_start;
    RETCODE       RETCODE       rc;
    DBINT         DBINT         rcint;
    char          char          bcphint[128];
    char          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("idxitmdl");

    InitString(i_name, I_NAME_LEN+1);
    InitString(i_data, I_DATA_LEN+1);

    sprintf(name, "%s..%s", aptr->database, "item");

    //rc = bcp_init(i_hdbc1, name, NULL, "logs\\item.err", DB_IN);
    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);
    }

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT4, 1);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN,
NULL, 0, 0, 3);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 4);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, I_DATA_LEN, NULL,
0, 0, 5);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    time_start = (TimeNow() / MILLI);

    item_rows_loaded = 0;

    for (i_id = 1; i_id <= max_items; i_id++)
    {
        i_im_id = RandomNumber(1L, 10000L);

        MakeAlphaString(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("idxitmdl");
}

//=====
//
// Function : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are
created
//
//=====

void LoadWarehouse()
{
    long          long          w_id;
    char          char          w_name[W_NAME_LEN+1];
    char          char          w_street_1[ADDRESS_LEN+1];
    char          char          w_street_2[ADDRESS_LEN+1];
    char          char          w_city[ADDRESS_LEN+1];
    char          char          w_state[STATE_LEN+1];
    char          char          w_zip[ZIP_LEN+1];
    double        double        w_tax;
    double        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("idxward");

InitString(w_name, W_NAME_LEN+1);
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, "warehouse.err");
sprintf(err_log_path, "%swarehouse%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 != SUCCEEDED)
    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 != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

time_start = (TimeNow() / MILLI);

warehouse_rows_loaded = 0;

for (w_id = (long)aptr->starting_warehouse; w_id <= (long)aptr->num_warehouses; w_id++)
{
    MakeAlphaString(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 != SUCCEEDED)
    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("idxward");

//stock_rows_loaded = 0;
stock_rows_loaded = 4294960000;
district_rows_loaded = 0;

District();
Stock();
}

//=====================================================
//
// Function : District
//
//=====================================================

void District()
{
    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("idxdisc1");

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 != SUCCEEDED)
    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 != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN,
NULL, 0, 0, 3);
if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN,
NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN,
NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0,
0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 10);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 11);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

d_ytd = 30000.0;
d_next_o_id = orders_per_district+1;
time_start = (TimeNow() / MILLI);

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++)
        {
            MakeAlphaString(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 != SUCCEEDED)
                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("idxdisc1");

return;
}

//=====
//
// Function : Stock
//
//=====

void Stock()
{
    long    s_i_id;
    long    s_w_id;
    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("idxstkcd");

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 != SUCCEEDED)
    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 != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
}

```

```

rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 2);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 3);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN,
NULL, 0, 0, 4);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN,
NULL, 0, 0, 5);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN,
NULL, 0, 0, 6);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN,
NULL, 0, 0, 7);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN,
NULL, 0, 0, 8);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN,
NULL, 0, 0, 9);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN,
NULL, 0, 0, 10);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN,
NULL, 0, 0, 11);

```

```

if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN,
NULL, 0, 0, 12);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN,
NULL, 0, 0, 13);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 14);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 15);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 16);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN,
NULL, 0, 0, 17);
if (rc != SUCCEEDED)
    HandleErrorDBC(w_hdbc1);

s_ytd = s_order_cnt = s_remote_cnt = 0;

time_start = (TimeNow() / MILLI);

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);

        }

    }

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("idxstkcd");

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("idxcust");
    // 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("idxhiscd");
}
}

```

```

}

// 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_hist,"%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("idxhiscd");
        }

        // 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\" > logs\nurand_load.log",
        /* if (aptr->starting_warehouse == 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;

        // fix to avoid ODBC float to numeric conversion
        // customer_buf[i].c_balance = 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);

            MakeAlphaString(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 =
%d\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;

        // Generate CUSTOMER and HISTORY data

        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;

        // fix to avoid ODBC float to numeric conversion
        // customer_buf[i].c_balance = -10.0;
        strcpy(customer_buf[i].c_balance,"-10.0");

        MakeAlphaString(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

        // Generate HISTORY data

```

```

        MakeAlphaString(12, 24, H_DATA_LEN,
customer_buff[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;

    // fix to avoid ODBC float to numeric conversion problem.

    // double          c_balance;
    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;

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0,
0, 4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL,
0, 0, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0,
6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0,
0, 7);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
8);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, 9);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, 10);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, 11);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0,
12);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN,
NULL, 0, SQLCHARACTER, 13);

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0,
14);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 15);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 16);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    // fix to avoid ODBC float to numeric conversion problem.

    // rc = bcp_bind(c_hdbc1, (BYTE *) &c_balance, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 17);
    // if (rc != SUCCEEDED)
    //     HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, 17);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, 18);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, 19);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 20);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, 500, NULL, 0, 0, 21);
    if (rc != SUCCEEDED)
        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;

// fix to avoid ODBC float to numeric conversion
problem.
// c_balance = customer_buf[i].c_balance;
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 != SUCCEEDED)
    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          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;

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN,
NULL, 0, SQLCHARACTER, 6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 7);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc2);

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);

```

```

    if (rc != SUCCEEDED)
        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 != SUCCEEDED)
            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("idxodcl");
}

// 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 != SUCCEEDED)
    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 != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
}

sprintf(name, "%s.%s", aptr->database, "new_order");

//
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 != SUCCEEDED)
    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 != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
}

sprintf(name, "%s.%s", aptr->database, "order_line");

//
DB_IN);
//
strcpy(err_log_path_ordl, aptr->log_path);
strcat(err_log_path_ordl, "ordline.err");
sprintf(err_log_path_ord, "%sordline%d.err", aptr->log_path, aptr->starting_warehouse);
rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl,
DB_IN);
if (rc != SUCCEEDED)
    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 != SUCCEEDED)
        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

    // 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 = %ld, w_id = %ld\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 = %d, w_id = %d\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
            }
        }
    }

```

```

        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
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0,
O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
}

```

```

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 7);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
    if (rc != SUCCEEDED)
        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 != SUCCEEDED)

            HandleErrorDBC(o_hdbc1);

        orders_rows_loaded++;
        CheckForCommit(o_hdbc1, o_hstmt1,
orders_rows_loaded, "orders", &orders_time_start->time_start);
    }

    // rcint = bcp_batch(o_hdbc1);
    // if (rcint < 0)
    //     HandleErrorDBC(o_hdbc1);

    if ((o_w_id == aprt->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("idxordc");

                // 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

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)

```

```

                HandleErrorDBC(o_hdbc2);

                for (i = first_new_order; i < last_new_order; i++)
                {
                    o_id = orders_buff[i].o_id;
                    o_d_id = orders_buff[i].o_d_id;
                    o_w_id = orders_buff[i].o_w_id;

                    rc = bcp_sendrow(o_hdbc2);
                    if (rc != SUCCEEDED)

                        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);
                }

                // rcint = bcp_batch(o_hdbc2);
                // if (rcint < 0)
                //     HandleErrorDBC(o_hdbc2);

                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("idxnodc");
                }
            }

//=====
//
// 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
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 1);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, 2);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 3);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 4);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, 5);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, 6);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);

```

```

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, 7);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA,
NULL, 0, SQLINT2, 8);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA,
NULL, 0, SQLFLT8, 9);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL,
0, 0, 10);
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);
    }
}

// rcint = bcp_batch(o_hdbc3);
// if (rcint < 0)
//     HandleErrorDBC(o_hdbc3);

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_diff = time_end - *time_start;

        printf("-> Loaded %ld rows into %s in %ld sec -
Total = %d (%.2f rps)\n",
                    aptr->batch,
                    table_name,
                    time_diff,
                    rows_loaded,
                    (float) aptr->batch /
                    (time_diff ? time_diff : 1L));

        *time_start = time_end;
    }

    return;
}

//=====
//
// Function : CheckForCommit_Big

```

```

//
//=====
void CheckForCommit_Big(HDBC hdbc,
                        hstmt,
                        rows_loaded,
                        *table_name,
                        long *time_start)
{
    long    time_end, time_diff;
    if ( !(fmod(rows_loaded,aptr->batch) ) )
    {
        time_end = (TimeNow() / MILLI);
        time_diff = time_end - *time_start;
        printf("> Loaded %ld rows into %s in %ld sec -
Total = %.0f (%.2f rps)\n",
        aptr->batch,
        table_name,
        time_diff,
        rows_loaded,
        (float) aptr->batch /
(time_diff ? time_diff : 1L));
        *time_start = time_end;
    }
    return;
}
//=====
//
// Function : OpenConnections
//
//=====
void OpenConnections()
{
    RETCODE    rc;
    char       szDriverString[300];

```

```

char
szDriverStringOut[1024];
SQLSMALLINT
int
int
int
cbDriverStringOut;
retrycnt;
done;

SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
&henv );
SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
(void*)SQL_OV_ODBC3, 0 );

// Open connections to SQL Server
// Connection 1
if (aptr->tables_all || aptr->table_item)
{
    retrycnt = 1;
    done = 0;
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&i_hdbc1);
    SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database );
    rc = SQLSetConnectOption ( i_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);
    while (!done)
    {
        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) )
{
    if (retrycnt <= aptr-
>retry_limit)
    {
        retrycnt++;
        Sleep(aptr->sleep_milli);
    } else
    {
        HandleErrorDBC(i_hdbc1);
        printf("TPC-C Loader aborted!\n");
        exit(9);
    } else
    done = 1;
}
} // of if item or all
// Connection 2
if (aptr->tables_all || aptr->table_warehouse)
{
    retrycnt = 1;
    done = 0;
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&w_hdbc1);
    SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);
while (!done)
{
    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) )
    {
        if (retrycnt <= aptr-
>retry_limit)
        {
            retrycnt++;
            Sleep(aptr->sleep_milli);
        } else
        {
            HandleErrorDBC(w_hdbc1);
            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
    } else
        done = 1;
}

```

```

    } // of if all or warehouse
// Connection 3
if (aptr->tables_all || aptr->table_customer)
{
    retrycnt = 1;
    done = 0;
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc1);
    SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);
    while (!done)
    {
        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) )
        {
            if (retrycnt <= aptr-
>retry_limit)

```

```

        {
            retrycnt++;
            Sleep(aptr->sleep_milli);
        } else
        {
            HandleErrorDBC(c_hdbc1);
            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
    } else
        done = 1;
}
// Connection 4
    retrycnt = 1;
    done = 0;
    SQLAllocHandle(SQL_HANDLE_DBC, henv ,
&c_hdbc2);
    SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP,
(void *)SQL_BCP_ON, SQL_IS_INTEGER );
    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);
    while (!done)
    {
        rc = SQLDriverConnect ( c_hdbc2,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,

```



```

        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
    if ( ( rc != SUCCEEDED ) &&
        (rc !=
SQL_SUCCESS_WITH_INFO) )
    {
        if (retrycnt <= aptr-
>retry_limit)
        {
            retrycnt++;
            Sleep(aptr->sleep_milli);
        } else
        {
            HandleErrorDBC(c_hdbc2);
            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
        done = 1;
    }

    } // of if all or customer

    if (aptr->tables_all || aptr->table_orders)
    {
        retrycnt = 1;
        done = 0;

        SQLAllocHandle(SQL_HANDLE_DBC, henv ,
        &o_hdbc1);
        SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP,
        (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        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 != SUCCEEDED)
            HandleErrorDBC(o_hdbc1);

        while (!done)
        {
            rc = SQLDriverConnect ( o_hdbc1,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if ( ( rc != SUCCEEDED ) &&
                (rc !=
SQL_SUCCESS_WITH_INFO) )
            {
                if (retrycnt <= aptr-
                >retry_limit)
                {
                    retrycnt++;
                    Sleep(aptr->sleep_milli);
                } else
                {
                    HandleErrorDBC(o_hdbc1);
                    printf("TPC-C Loader aborted!\n");
                    exit(9);
                }
                done = 1;
            }
        }

        // Connection 6

        retrycnt = 1;

```

```

        done = 0;
        SQLAllocHandle(SQL_HANDLE_DBC, henv ,
        &o_hdbc2);
        SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP,
        (void *)SQL_BCP_ON, SQL_IS_INTEGER );
        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 != SUCCEEDED)
            HandleErrorDBC(o_hdbc2);

        while (!done)
        {
            rc = SQLDriverConnect ( o_hdbc2,
            NULL,
            (SQLCHAR*)&szDriverString[0] ,
            SQL_NTS,
            (SQLCHAR*)&szDriverStringOut[0],
            sizeof(szDriverStringOut),
            &cbDriverStringOut,
            SQL_DRIVER_NOPROMPT );
            if ( ( rc != SUCCEEDED ) &&
                (rc !=
SQL_SUCCESS_WITH_INFO) )
            {
                if (retrycnt <= aptr-
                >retry_limit)
                {
                    retrycnt++;
                    Sleep(aptr->sleep_milli);
                } else

```

```

        {
        HandleErrorDBC(o_hdbc2);
        printf("TPC-C Loader aborted!\n");
        } else
        {
            done = 1;
        }
    }
    // Connection 7
    retrycnt = 1;
    done = 0;
    SQLAllocHandle(SQL_HANDLE_DBC, henv,
    &o_hdbc3);
    SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP,
    (void *)SQL_BCP_ON, SQL_IS_INTEGER);
    sprintf( szDriverString, "DRIVER={SQL
    Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
    apr->server,
    apr->user,
    apr->password,
    apr->database );
    rc = SQLSetConnectOption (o_hdbc3,
    SQL_PACKET_SIZE, apr->pack_size);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    while (!done)
    {
        rc = SQLDriverConnect ( o_hdbc3,
        NULL,
        (SQLCHAR*)&szDriverString[0],
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
    }
}

```

```

    if ( (rc != SUCCEEDED) &&
    (rc !=
    SQL_SUCCESS_WITH_INFO) )
    {
        if (retrycnt <= apr-
        >retry_limit)
        {
            retrycnt++;
            Sleep(apr->sleep_milli);
        } else
        {
            HandleErrorDBC(o_hdbc3);
            printf("TPC-C Loader aborted!\n");
            exit(9);
        }
    } else
    {
        done = 1;
    }
} //
//=====
//
// 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",
    apr->server,
    apr->user,
    apr->password,
    apr-
    >index_script_path,
    index_script,
    apr->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 )
    {
        if (totaldberrors >= MAXSQLERRORS)
        {
            printf(">>>>>Max SQL errors of
            %d exceeded, terminating tpcldr<<<<<<\n",totaldberrors);
            exit(9);
        }
        sprintf( szLastError, "%s", Msg );
        _strtime(timebuf);
        _strdate(datebuf);
        printf( "[%s : %s] %s\nSQLState: %s\n",
        datebuf, timebuf, szLastError, SqlState);
        //
        strcpy(err_log_path,apr->log_path);
        strcat(err_log_path,"tpcldr.err");
        printf(err_log_path,"%stpcldr%d.err",apr->log_path,apr-
        >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 (totaldberrors >= MAXSQLERRORS)
        {
            printf(">>>>>>Max SQL errors of
%d exceeded, terminating tpccldr<<<<<<<\n",totaldberrors);
            exit(9);
        }
        totaldberrors++;

        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);
//      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 : 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;
}

```

```

//=====
//
// Function : CheckDataBase
//
//=====

void CheckDataBase()
{
    RETCODE rc;

    char          szDriverString[300];
    char          szDriverStringOut[1024];
    char          TablesBitMap[9] =
{"000000000"};
    int          i,
    ExitFlag;

    SQLSMALLINT          cbDriverStringOut;
    SQLCHAR              TabName[10];

#ifdef _WIN64
    SQLINTEGER          TabCount;
    SQLLEN              TabNameInd,
    TabCountInd;
#else
    SQLINTEGER          TabNameInd, TabCount,
    TabCountInd;
#endif

    ExitFlag = 0;

    SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE,
    &henv );

    SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION,
    (void*)SQL_OV_ODBC3, 0 );

    SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

    SQLSetConnectAttr(v_hdbc, SQL_COPT_SS_BCP, (void
*)SQL_BCP_ON, SQL_IS_INTEGER );

    // Open connection to SQL Server

    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
        aptr->server,

```

```

aptr->user,
aptr->password,
aptr->database );
rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE,
(SQLPOINTER)aptr->pack_size, SQL_IS_UIINTEGER );
if (rc != SQL_SUCCESS)
    HandleErrorDBC(v_hdbc);
rc = SQLDriverConnect ( v_hdbc,
                        NULL,
(SQLCHAR*)&szDriverString[0] ,
SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
&v_hstmt);
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
// if the rc is SQL_ERROR, the the TPCC database probably
does not exist
if (rc == SQL_ERROR)
{
    printf("The database TPCC does not appear to
exist!\n");
    printf("\nCheck LOGS\ directory for database
creation errors.\n");
    // cleanup database connections and handles
SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
SQLDisconnect(v_hdbc);
SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
// since there is not a database, exit back to
SETUP.CMD
    exit(1);
}
if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt)
!= SQL_SUCCESS )
    HandleErrorDBC(v_hdbc);
if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0,
&TabCountInd) != SQL_SUCCESS )

```

```

    HandleErrorSTMT(v_hstmt);
// count the number of user tables from sysobjects
rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects
where xtype = '\U'", SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);
if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);
// if the number of tables is less than 9, select all the user
tables in TPCC
if (TabCount != 9)
{
    SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
    SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc ,
&v_hstmt);
if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR,
&TabName, sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
    HandleErrorSTMT(v_hstmt);
// select the list of user tables into a result set
rc = SQLExecDirect(v_hstmt, "select * from
sysobjects where xtype = '\U'", SQL_NTS);
if ((rc != SQL_SUCCESS) && (rc !=
SQL_SUCCESS_WITH_INFO))
    HandleErrorSTMT(v_hstmt);
// go through the result set and set the bitmap for
each found table
// set the bitmap to '1' if the table name is found
while ((rc = SQLFetch(v_hstmt)) !=
SQL_NO_DATA)
{
    switch( TabName[0] )
    {
        case 'w':
            TablesBitMap[0] = '1';
            break;
        case 'd':
            TablesBitMap[1] = '1';
            break;
        case 'c':
            TablesBitMap[2] = '1';
            break;
        case 'h':

```

```

            TablesBitMap[3] = '1';
            break;
        case 'n':
            TablesBitMap[4] = '1';
            break;
        case 'o':
            if (TabName[5] = 's')
                if (TabName[5] = '_')
                    break;
            case 'i':
                TablesBitMap[7] = '1';
                break;
            case 's':
                TablesBitMap[8] = '1';
                break;
        }
    }
// a '0' ExitFlag means do NOT exit the loader
early, a '1' means exit the loader early
ExitFlag = 0;
// iterate through the bitmap to display which
table(s) is actually missing
for (i = 0; i <= 8; i++)
{
    switch(i)
    {
        case 0:
            if (TablesBitMap[i] ==
'0')
            {
                printf("The Warehouse table is missing or damaged.\n");
                ExitFlag
= 1;
            }
            case 1:
                if (TablesBitMap[i] ==
'0')
                {
                    printf("The District table is missing or damaged.\n");
                    ExitFlag
= 1;
                }
    }
}

```

```

    case 2:
        break;
    case 2:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The Customer table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    case 3:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The History table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    case 4:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The New_Order table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    case 5:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The Orders table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    case 6:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The Order_Line table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;

```

```

    case 7:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The Item table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    case 8:
        if (TablesBitMap[i] ==
'0')
        {
            printf("The Stock table is missing or damaged.\n");
            ExitFlag
= 1;
        }
        break;
    }
    // if one or more tables are missing, display
    message and exit the loader
    if (ExitFlag = 1)
    {
        printf("\nExiting TPC-C Loader!\n");
        printf("\nCheck LOGS\\ directory for
database\n");
        printf("or table creation errors.\n");
        // cleanup database connections and
handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC,
v_hdbc);
    }
    exit(1);
}
// cleanup database connections and handles
SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
SQLDisconnect(v_hdbc);
SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);
return;
}

```

### time.c

```

//      File:          TIME.C
//      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("[%d]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
//
//      Ver. 4.42
//
//      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("[%d]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("[%d]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("[%d]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 <%d> out
            of range (0,999)\n", num);
            exit(-1);
        }
    }

#ifdef DEBUG
    printf("[%d]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
            (int) GetCurrentThreadId(), num,
            num/100, (num/10)%10, num%10);
    printf("[%d]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:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-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("[%d]DBG: Entering MakeAlphaString()\n", (int)
    GetCurrentThreadId());
#endif

    len= RandomNumber(x, y);
    for (i=0; i<len; i++)
    {
        cc = chArray[RandomNumber(0, chArrayMax)];
        str[i] = cc;
    }

    //if ( len < z )
    //    memset(str+len, ' ', z - len);
    str[len] = 0;
}

```

```

return len;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====
int MakeOriginalAlphaString(int x,
                           int y,
                           int z,
                           char *str,
                           int percent)
{
    int len;
    int val;
    int start;

#ifdef DEBUG
    printf("[%d]DBG: Entering MakeOriginalAlphaString()\n", (int)
GetCurrentThreadId());
#endif

    // verify percentage is valid
    if ((percent < 0) || (percent > 100))
    {
        printf("MakeOriginalAlphaString: Invalid
percentage: %d\n", percent);
        exit(-1);
    }

    // verify string is at least 8 chars in length
    if ((x + y) <= 8)
    {
        printf("MakeOriginalAlphaString: string length
must be >= 8\n");
        exit(-1);
    }

    // Make Alpha String
    len = MakeAlphaString(x,y, z, str);

    val = RandomNumber(1,100);

```

```

if (val <= percent)
{
    start = RandomNumber(0, len - 8);
    strcpy(str + start, "ORIGINAL", 8);
}

#ifdef DEBUG
    printf("[%d]DBG: MakeOriginalAlphaString: : %s\n",
           (int) GetCurrentThreadId(), str);
#endif

return strlen(str);
}

//=====
//
// Function name: MakeNumberString
//
//=====
int MakeNumberString(int x, int y, int z, char *str)
{
    char tmp[16];

    //MakeNumberString is always called MakeZipNumberString(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("[%d]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 M            2147483647
#define Q            127773 /* M div A */
#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("[%d]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("[%d]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("[%d]DBG: RandomNumber between %ld & %ld ==> %ld\n",
(int)
GetCurrentThreadId(), lower, upper, rand_num);
#endif

return rand_num;
}

#if 0

```

```

//Original code pgd 08/13/96
long RandomNumber(long lower,
long upper)
{
long rand_num;

#ifdef DEBUG
printf("[%d]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("[%d]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("[%d]DBG: Entering NURand()...\n", (int) GetCurrentThreadId());
#endif

```

```

rand_num = (((RandomNumber(0,iConst) | RandomNumber(x,y)) + C) %
(y-x+1))+x;

#ifdef DEBUG
printf("[%d]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("[%d]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 through */
            GetArgsLoaderUsage();
            break;

        case 'D':
            pargs->database = ptr+2;

        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 = 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;
            if ( strcmp(ptr+2,"warehouse") == 0 )
                pargs->table_warehouse = TRUE;
            if ( strcmp(ptr+2,"customer") == 0 )
                pargs->table_customer = TRUE;
            if ( strcmp(ptr+2,"orders") == 0 )
                pargs->table_orders = TRUE;
            break;

        default:
            printf("\nUnrecognized command");
            GetArgsLoaderUsage();
            exit(1);
            break;
    }
}

else
{
    printf("\nUnrecognized command");
    GetArgsLoaderUsage();
    exit(1);
}

case 'f':
    pargs->loader_res_file = ptr+2;
    break;

case 'l':
    pargs->log_path = ptr+2;
    break;

case 'p':
    pargs->pack_size = ptr+2;
    break;

case 'i':
    pargs->build_index = ptr+2;
    break;

case 'o':
    pargs->index_order = ptr+2;
    break;

case 'c':
    pargs->scale_down = ptr+2;
    break;

case 'd':
    pargs->index_script_path = ptr+2;
    break;

case 'R':
    pargs->retry_limit = 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("[%d]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                        %d\n", (long)
BATCH);

```

```

        printf("-p TDS packet size                %d\n",
(long) DEF_LD_PACKET_SIZE);
        printf("-l Loader Results Output Filename    %s\n",
LOADER_RES_FILE);
        printf("-s Starting Warehouse                %d\n",
(long) DEF_STARTING_WAREHOUSE);
        printf("-i Build Option (data = 0, data and index = 1)
%d\n", (long) BUILD_INDEX);
        printf("-o Cluster Index Build Order (before = 1, after = 0)
%d\n", (long) INDEX_ORDER);
        printf("-R Retry limit. # of times to retry open    %d\n",
(long) RETRYLIMIT);
        printf("-r Retry sleep time in milliseconds.    %d\n",
(long) SLEPMILLI);
        printf("-c Build Scaled Database (normal = 0, tiny = 1)
%d\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. 4.20
-- Copyright Microsoft, 1999
-- Purpose: Sets database options for data load

```

```

use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'auto create statistics',true
exec sp_dboption tpcc,'auto update statistics',true
go

use tpcc
go

alter database tpcc set recovery bulk_logged
go

alter database tpcc set torn_page_detection off
go

checkpoint
go

```

### DBOPT2 . SQL

```

-- File: DBOPT2_SHILOH.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.10a
-- Copyright Microsoft, 1999
-- Purpose: Resets database options after data load (For Shiloh)

```

```

use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint

```

```

go

alter database tpcc set recovery full
go

sp_configure allow,1
go

reconfigure with override
go

/*          */
/* Set option values for user-defined indexes */
/*          */

sp_indexoption 'customer', 'DisallowPageLocks', TRUE
go
sp_indexoption      'district', 'DisallowPageLocks', TRUE
go
sp_indexoption      'warehouse', 'DisallowPageLocks', TRUE
go
sp_indexoption      'stock', 'DisallowPageLocks', TRUE
go
sp_indexoption      'order_line', 'DisallowRowLocks', TRUE
go
sp_indexoption      'orders', 'DisallowRowLocks', TRUE
go
sp_indexoption      'new_order', 'DisallowRowLocks', TRUE
go
sp_indexoption      'item', 'DisallowRowLocks',
TRUE
go
sp_indexoption      'item', 'DisallowPageLocks',
TRUE
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("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,0
go

reconfigure with override
go

exec sp_dboption tpcc, 'auto update statistics', FALSE
exec sp_dboption tpcc, 'auto create statistics', FALSE
go

exec sp_tableoption "district", "pintable",true
exec sp_tableoption "warehouse","pintable",true
exec sp_tableoption "new_order","pintable",true
exec sp_tableoption "item", "pintable",true
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

```

```

( FILENAME = "g:\tpcc9800\root",
  SIZE = 12MB,
  FILEGROWTH = 0),

FILEGROUP cs_fg
(NAME=cs1, FILENAME="g:\mnt\cs\1\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs2, FILENAME="g:\mnt\cs\2\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs3, FILENAME="g:\mnt\cs\3\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs4, FILENAME="g:\mnt\cs\4\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs5, FILENAME="g:\mnt\cs\5\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs6, FILENAME="g:\mnt\cs\6\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs7, FILENAME="g:\mnt\cs\7\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs8, FILENAME="g:\mnt\cs\8\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs9, FILENAME="g:\mnt\cs\9\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs10, FILENAME="g:\mnt\cs\10\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs11, FILENAME="g:\mnt\cs\11\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs12, FILENAME="g:\mnt\cs\12\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs13, FILENAME="g:\mnt\cs\13\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs14, FILENAME="g:\mnt\cs\14\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs15, FILENAME="g:\mnt\cs\15\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs16, FILENAME="g:\mnt\cs\16\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs17, FILENAME="g:\mnt\cs\17\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs18, FILENAME="g:\mnt\cs\18\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs19, FILENAME="g:\mnt\cs\19\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs20, FILENAME="g:\mnt\cs\20\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs21, FILENAME="g:\mnt\cs\21\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs22, FILENAME="g:\mnt\cs\22\",
SIZE=94236MB,FILEGROWTH=0),
(NAME=cs23, FILENAME="g:\mnt\cs\23\",
SIZE=94236MB,FILEGROWTH=0),

```



```

(NAME=misc40, FILENAME="g:\mnt\misc\40\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc41, FILENAME="g:\mnt\misc\41\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc42, FILENAME="g:\mnt\misc\42\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc43, FILENAME="g:\mnt\misc\43\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc44, FILENAME="g:\mnt\misc\44\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc45, FILENAME="g:\mnt\misc\45\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc46, FILENAME="g:\mnt\misc\46\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc47, FILENAME="g:\mnt\misc\47\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc48, FILENAME="g:\mnt\misc\48\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc49, FILENAME="g:\mnt\misc\49\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc50, FILENAME="g:\mnt\misc\50\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc51, FILENAME="g:\mnt\misc\51\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc52, FILENAME="g:\mnt\misc\52\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc53, FILENAME="g:\mnt\misc\53\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc54, FILENAME="g:\mnt\misc\54\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc55, FILENAME="g:\mnt\misc\55\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc56, FILENAME="g:\mnt\misc\56\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc57, FILENAME="g:\mnt\misc\57\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc58, FILENAME="g:\mnt\misc\58\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc59, FILENAME="g:\mnt\misc\59\",
SIZE=52227MB,FILEGROWTH=0),
(NAME=misc60, FILENAME="g:\mnt\misc\60\",
SIZE=52227MB,FILEGROWTH=0),

FILEGROUP load_fg
(NAME=load1, FILENAME="g:\mnt\load\1\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load2, FILENAME="g:\mnt\load\2\",
SIZE=113000MB,FILEGROWTH=0),

```

```

(NAME=load3, FILENAME="g:\mnt\load\3\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load4, FILENAME="g:\mnt\load\4\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load5, FILENAME="g:\mnt\load\5\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load6, FILENAME="g:\mnt\load\6\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load7, FILENAME="g:\mnt\load\7\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load8, FILENAME="g:\mnt\load\8\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load9, FILENAME="g:\mnt\load\9\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load10, FILENAME="g:\mnt\load\10\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load11, FILENAME="g:\mnt\load\11\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load12, FILENAME="g:\mnt\load\12\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load13, FILENAME="g:\mnt\load\13\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load14, FILENAME="g:\mnt\load\14\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load15, FILENAME="g:\mnt\load\15\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load16, FILENAME="g:\mnt\load\16\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load17, FILENAME="g:\mnt\load\17\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load18, FILENAME="g:\mnt\load\18\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load19, FILENAME="g:\mnt\load\19\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load20, FILENAME="g:\mnt\load\20\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load21, FILENAME="g:\mnt\load\21\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load22, FILENAME="g:\mnt\load\22\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load23, FILENAME="g:\mnt\load\23\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load24, FILENAME="g:\mnt\load\24\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load25, FILENAME="g:\mnt\load\25\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load26, FILENAME="g:\mnt\load\26\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load27, FILENAME="g:\mnt\load\27\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load28, FILENAME="g:\mnt\load\28\",
SIZE=113000MB,FILEGROWTH=0),

```

```

(NAME=load29, FILENAME="g:\mnt\load\29\",
SIZE=113000MB,FILEGROWTH=0),
(NAME=load30, FILENAME="g:\mnt\load\30\",
SIZE=113000MB,FILEGROWTH=0)

```

```

LOG ON
(
NAME = tpcc_log,
FILENAME = "L:",
SIZE = 200000MB,
FILEGROWTH = 0)

```

#### Createtables.sql

```

--
-- HPTPC Setup Kit
-- Drop and Create Tables
-- Tables are initially created in the load_fg filegroup
--

use tpcc
go

SET ANSI_NULL_DFLT_OFF ON
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

```

<pre> 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_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),     w_tax              numeric(4,4),     w_ytd              numeric(12,2) ) on load_fg go  create table district (     d_id                tinyint,     d_w_id              int,     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),     d_tax              numeric(4,4),     d_ytd              numeric(12,2),     d_next_o_id        int ) on load_fg go  create table customer (     c_id                int, </pre>	<pre> e_w_id                tinyint, c_first              char(16), c_middle             char(2), c_last              char(16), 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        numeric(12,2), c_discount           numeric(4,4), c_balance           numeric(12,2), c_ytd_payment       numeric(12,2), c_payment_cnt       smallint, c_delivery_cnt       smallint, c_data              char(500) ) on load_fg go  create table history (     h_c_id            int,     h_c_d_id          tinyint,     h_c_w_id          int,     h_d_id            int,     h_w_id            tinyint,     h_date            datetime,     h_amount          numeric(6,2),     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_entry_d         datetime, </pre>	<pre> o_all_local         tinyint, ) 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_supply_w_id   int,     ol_delivery_d     datetime,     ol_quantity       smallint,     ol_amount         numeric(6,2),     ol_dist_info     char(24) ) on load_fg go  create table item (     i_id              int,     i_im_id           int,     i_name            char(24),     i_price           numeric(5,2),     i_data            char(50) ) on load_fg go  create table stock (     s_i_id            int,     s_w_id            int,     s_quantity        smallint,     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),     s_ytd             int,     s_order_cnt       smallint,     s_remote_cnt     smallint,     s_data            char(50) </pre>
--	---	--

```

) on load_fg
go

sp_tableoption 'warehouse','table lock on bulk load','on'
go
sp_tableoption 'district','table lock on bulk load','on'
go
sp_tableoption 'customer','table lock on bulk load','on'
go
sp_tableoption 'history','table lock on bulk load','on'
go
sp_tableoption 'new_order','table lock on bulk load','on'
go
sp_tableoption 'orders','table lock on bulk load','on'
go
sp_tableoption 'order_line','table lock on bulk load','on'
go
sp_tableoption 'stock','table lock on bulk load','on'
go

```

#### newcreateCustomerindexes.sql

```

--
-- HP TPCC Setup
-- Create Indexes for Customer
-- and move the table from LoadFg to CustomerStockFg
--
use tpcc
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 unique 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

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

```

```
--
```

#### 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

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)
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

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\_32-neword\_old.sql

```

-----
-- File: NEWORD.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.42 --
-- Copyright Microsoft, 2003 --
--
-- - Creates new order transaction stored procedure --
-- - Interface Level: 4.10.000 --
-----

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_neworder' )
    drop procedure tpcc_neworder
go

create proc tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,

```

```

int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

```

```

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),
        @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,

```

```

        @i_id2 int,
        @i_id3 int,
        @i_id4 int,
        @i_id5 int,
        @i_id6 int,
        @i_id7 int,
        @i_id8 int,
        @i_id9 int,
        @i_id10 int,
        @i_id11 int,
        @i_id12 int,
        @i_id13 int,
        @i_id14 int,
        @i_id15 int,
        @c_id_local int
begin
begin transaction n
-----
-- get district tax and next available 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
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
and 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,
@s_d_id,
@s_w_id,
@s_li_no,
@s_li_id,
@s_li_s_w_id,
'dec 31, 1899',
@s_li_qty,
@s_i_price * @li_qty,
@s_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))

```

```

end,
                                then 'B' else 'G'
                                @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_entry_d,
                                                                0,
                                                                @o_ol_cnt,
                                                                @o_all_local)
-----
-- insert corresponding row into new-order table
-----
                                insert into new_order values ( @o_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
                                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,

```

```

@w_id)
                                int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
                                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      numeric(4,4),
                                @d_tax      numeric(4,4),
                                @c_last     char(16),
                                @c_credit   char(2),
                                @c_discount numeric(4,4),
                                @i_price    numeric(5,2),
                                @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 available 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
                when 2 then
                when 3 then
                when 4 then
                when 5 then
                when 6 then
                when 7 then
                when 8 then
                when 9 then
                when 10 then
                when 11 then
                when 12 then
                when 13 then
                when 14 then
                when 15 then

end,

@ol_qty9
@ol_qty10

@ol_qty11

@ol_qty12

@ol_qty13

@ol_qty14

@ol_qty15

-----
-- 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
                when 1
                when 2
                when 3
                when 4
                when 5
                when 9 then
                when 10 then
                when 11 then
                when 12 then
                when 13 then
                when 14 then
                when 15 then
                end

end

```

```

then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

and         where    s_i_id
@li_s_w_id      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,

                        @li_s_w_id,

                        'dec 31, 1899',

                        @li_qty,

                        @i_price * @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,"0,0,0
                                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
                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_entry_d,

                        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
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,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare    @w_tax      numeric(4,4),
           @d_tax      numeric(4,4),
           @c_last     char(16),
           @c_credit   char(2),
           @c_discount numeric(4,4),
           @i_price    numeric(5,2),
           @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,

```

```

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@li_id @commit_flag tinyint,
@li_s_w_id int,
@li_qty smallint,
@ol_number int,
@c_id_local int

begin
begin transaction n
-----
-- get district tax and next available 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 l_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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

```

```

when 15 then @i_id15
@s_w_id = case @li_no
         when 1 then
         when 2 then
         when 3 then
         when 4 then
         when 5 then
         when 6 then
         when 7 then
         when 8 then
         when 9 then
         when 10 then
         when 11 then
         when 12 then
         when 13 then
         when 14 then
         when 15 then
end,
@li_qty = case @li_no
         when 1 then
         when 2 then
         when 3 then
         when 4 then
         when 5 then
         when 6 then
         when 7 then

```

```

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

then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

where       s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)

-----
-- send line-item data to client
-----
select      @i_name,
            @s_quantity,

when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=

-----
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_entry_d,
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
    rollback transaction n
-----
-- 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
if exists ( select name from sysobjects where name = 'sp2tpcc_neworder' )
    drop procedure sp2tpcc_neworder

go

create proc sp2tpcc_neworder
int,
tinyint,
int,
@o_ol_cnt tinyint,

```

```

@o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare      @w_tax      numeric(4,4),
            @d_tax      numeric(4,4),
            @c_last     char(16),
            @c_credit   char(2),
            @c_discount numeric(4,4),
            @i_price    numeric(5,2),
            @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,

```

```

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

@li_s_w_id int,
@li_qty smallint,
@ol_number int,
@c_id_local int

begin
begin transaction n
-----
-- get district tax and next available 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
  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,
  @li_s_w_id,
  'dec 31, 1899',
  @li_qty,
  @i_price * @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_entry_d,
                            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
            rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
if exists ( select name from sysobjects where name = 'sp3tpcc_neworder' )
drop procedure sp3tpcc_neworder
go
create proc sp3tpcc_neworder
int,
tinyint,
int,
@o_ol_cnt tinyint,

```

```

            @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
            @i_id1
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
            @i_id2
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
            @i_id3
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
            @i_id4
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
            @i_id5
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
            @i_id6
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
            @i_id7
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
            @i_id8
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
            @i_id9
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
            @i_id10
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
            @i_id11
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
            @i_id12
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
            @i_id13
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
            @i_id14
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0,
            @i_id15
as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),
        @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 available 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

                    @ol_qty1
                    @ol_qty2
                    @ol_qty3
                    @ol_qty4
                    @ol_qty5
                    @ol_qty6
                    @ol_qty7

                    when 8 then
                    when 9 then
                    when 10 then
                    when 11 then
                    when 12 then
                    when 13 then
                    when 14 then
                    when 15 then
                    end,
                    @li_qty = case @li_no
                    when 1 then
                    when 2 then
                    when 3 then
                    when 4 then
                    when 5 then
                    when 6 then
                    when 7 then
                    when 8 then
                    when 9 then
                    when 10 then
                    when 11 then
                    when 12 then
                    when 13 then
                    when 14 then
                    when 15 then
                    end,
                    @ol_qty8
                    @ol_qty9
                    @ol_qty10
                    @ol_qty11
                    @ol_qty12
                    @ol_qty13
                    @ol_qty14
                    @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 = @i_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
                    when 1
                    when 2
                    when 3
                    when 4
                    when 8 then
                    when 9 then
                    when 10 then
                    when 11 then
                    when 12 then
                    when 13 then
                    when 14 then
                    when 15 then
                    end
                    end

```

```

then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

and          where   s_i_id
              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,
          @li_s_w_id,
          'dec 31, 1899',
          @li_qty,
          @i_price * @li_qty,
          @s_dist)

-----
-- send line-item data to client
-----
          select      @i_name,

```

```

when 5
when 6
when 7
when 8
when 9
when
end
when
end
else
begin
-----
-- no item (or stock) found - triggers rollback condition
-----
          select "0", "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,

          @c_id_local,
          @o_entry_d,

          @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
if exists ( select name from sysobjects where name = 'sp4tpcc_neworder' )
drop procedure sp4tpcc_neworder
go

create proc sp4tpcc_neworder

int,
tinyint,
int,

@o_ol_cnt tinyint,

```

```

        @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

```

```

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),
        @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,

```

```

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

        @li_id @commit_flag tinyint,
        @li_s_w_id int,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int
begin
begin transaction n
-----
-- get district tax and next available 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,
                @ol_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
when 2
when 3
when 4
when 8 then
then s_dist_05
then s_dist_06
when 10 then
then s_dist_07
when 11 then
then s_dist_08
when 12 then
then s_dist_09
10 then s_dist_10
where       s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)

-----
-- send line-item data to client
-----
select      @i_name,
            @s_quantity,
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=
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_entry_d,
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
    rollback transaction n
-----
-- 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
if exists ( select name from sysobjects where name = 'sp5tpcc_neworder' )
    drop procedure sp5tpcc_neworder

go

create proc sp5tpcc_neworder
int,
tinyint,
int,
@o_ol_cnt tinyint,

```

```

@o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare      @w_tax      numeric(4,4),
            @d_tax      numeric(4,4),
            @c_last     char(16),
            @c_credit   char(2),
            @c_discount numeric(4,4),
            @i_price    numeric(5,2),
            @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,

```

```

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

@li_s_w_id int,
@li_qty     smallint,
@ol_number int,
@c_id_local int

begin
begin transaction n
-----
-- get district tax and next available 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
  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,
  @li_s_w_id,
  'dec 31, 1899',
  @li_qty,
  @i_price * @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_entry_d,
                                                                0,
                                                                tinyint,
                                                                @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
                                        rollback transaction n
-----
-- 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
if exists ( select name from sysobjects where name = 'sp6tpcc_neworder' )
drop procedure sp6tpcc_neworder
go
create proc sp6tpcc_neworder
int,
                                @w_id
tinyint,
                                @d_id
int,
                                @c_id
                                @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 numeric(4,4),
@d_tax numeric(4,4),
@c_last char(16),
@c_credit char(2),
@c_discount numeric(4,4),
@i_price numeric(5,2),
@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 available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7

```

```

    when 15 then @i_id15
@s_w_id = case @li_no
            when 1 then
            when 2 then
            when 3 then
            when 4 then
            when 5 then
            when 6 then
            when 7 then
            when 8 then
            when 9 then
            when 10 then
            when 11 then
            when 12 then
            when 13 then
            when 14 then
            when 15 then
    end,
@li_qty = case @li_no
            when 1 then
            when 2 then
            when 3 then
            when 4 then
            when 5 then
            when 6 then
            when 7 then

```

```

            when 8 then
            when 9 then
            when 10 then
            when 11 then
            when 12 then
            when 13 then
            when 14 then
            when 15 then
    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
    when 1
    when 2
    when 3
    when 4

```

```

then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

and
@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,
@li_s_w_id,
@d_id,
@w_id,
@li_no,
@li_id,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)

-----
-- send line-item data to client
-----
select @i_name,

when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=

-----
-- 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,
@o_entry_d,
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
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
if exists ( select name from sysobjects where name = 'sp7tpcc_neworder' )
drop procedure sp7tpcc_neworder
go

create proc sp7tpcc_neworder
int,
tinyint,
int,
@o_ol_cnt tinyint,
@w_id
@d_id
@c_id

```

```

        @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

```

```

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),
        @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,

```

```

@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

        @li_id @commit_flag tinyint,
        @li_s_w_id int,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int
begin
begin transaction n
-----
-- get district tax and next available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7

```

```

when 15 then @i_id15
@li_s_w_id = case @li_no
    when 1 then
    when 2 then
    when 3 then
    when 4 then
    when 5 then
    when 6 then
    when 7 then
    when 8 then
    when 9 then
    when 10 then
    when 11 then
    when 12 then
    when 13 then
    when 14 then
    when 15 then
end,
@li_qty = case @li_no
    when 1 then
    when 2 then
    when 3 then
    when 4 then
    when 5 then
    when 6 then
    when 7 then

```

```

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
10 then s_dist_10
where       s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)
when 1
when 2
when 3
when 4
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_entry_d,
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
    rollback transaction n

-----
-- all that work for nuthn!!!
-----
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

if exists ( select name from sysobjects where name = 'sp8tpcc_neworder' )
    drop procedure sp8tpcc_neworder
go
create proc sp8tpcc_neworder
int,
tinyint,

```

```

int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare      @w_tax numeric(4,4),
            @d_tax numeric(4,4),
            @c_last char(16),
            @c_credit char(2),
            @c_discount numeric(4,4),
            @i_price numeric(5,2),
            @i_name char(24),
            @i_data char(50),
            @o_entry_d datetime,
            @remote_flag int,
            @s_quantity smallint,
            @s_data char(50),

```

```

@c_id
    @s_dist@li_nochar(24)nt,
    @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 available 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
@s_w_id1      when 1 then
@s_w_id2      when 2 then
@s_w_id3      when 3 then
@s_w_id4      when 4 then
@s_w_id5      when 5 then
@s_w_id6      when 6 then
@s_w_id7      when 7 then
@s_w_id8      when 8 then
@s_w_id9      when 9 then
@s_w_id10     when 10 then
@s_w_id11     when 11 then
@s_w_id12     when 12 then
@s_w_id13     when 13 then
@s_w_id14     when 14 then
@s_w_id15     when 15 then
end,
@li_qty = case @li_no
@ol_qty1      when 1 then
@ol_qty2      when 2 then
@ol_qty3      when 3 then
@ol_qty4      when 4 then
@ol_qty5      when 5 then
@ol_qty6      when 6 then
@ol_qty7      when 7 then
@ol_qty8      when 8 then
@ol_qty9      when 9 then
@ol_qty10     when 10 then
@ol_qty11     when 11 then
@ol_qty12     when 12 then
@ol_qty13     when 13 then
@ol_qty14     when 14 then
@ol_qty15     when 15 then
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
and         where      s_i_id = @li_id
            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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @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_entry_d,
                            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
            rollback transaction n
-----
-- 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
if exists ( select name from sysobjects where name = 'sp9tpcc_neworder' )
            drop procedure sp9tpcc_neworder
go
create proc sp9tpcc_neworder
            @w_id
            int,
            tinyint,
            int,
            @o_ol_cnt tinyint,
            @o_all_local tinyint,
            int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
            int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
            int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
            int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
            int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
            int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
            int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
            int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
            int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
            int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
            int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
            int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
            int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
            int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
            int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
as
declare      @w_tax   numeric(4,4),
            @d_tax   numeric(4,4),
            @c_last  char(16),
            @c_credit char(2),
            @c_discount numeric(4,4),
            @i_price numeric(5,2),
            @i_name  char(24),
            @i_data  char(50),
            @o_entry_d datetime,
            @remote_flag int,
            @d_id
            @c_id
            @i_id1
            @i_id2
            @i_id3
            @i_id4
            @i_id5
            @i_id6
            @i_id7
            @i_id8
            @i_id9
            @i_id10
            @i_id11
            @i_id12
            @i_id13
            @i_id14
            @i_id15

```



```

@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 available 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
    when 2 then
    when 3 then
    when 4 then
    when 5 then
    when 6 then
    when 7 then
    when 8 then
    when 9 then
    when 10 then
    when 11 then
    when 12 then
    when 13 then
    when 14 then
    when 15 then
end,
@li_qty = case @li_no
    when 1 then
    when 2 then
    when 3 then
    when 4 then
    when 5 then
    when 6 then
    when 7 then
    when 8 then
    when 9 then
    when 10 then
    when 11 then
    when 12 then
    when 13 then
    when 14 then
    when 15 then
end,
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15

-----
-- 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
    when 1
then s_dist_01

```

```

then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

and
@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,

@li_s_w_id,

'dec 31, 1899',

@li_qty,

@i_price * @li_qty,

when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
=
=

-----
-- 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
from
where
c_id
c_w_id
c_d_id

@c_last = c_last,
@c_discount = c_discount,
@c_credit = c_credit,
@c_id_local = c_id
customer WITH (repeatableread)
= @c_id and
= @w_id and
= @d_id

-----
-- insert fresh row into orders table
-----
insert into orders values ( @o_id,

@o_entry_d,
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
from
where

@w_tax = w_tax
warehouse WITH (repeatableread)
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
if exists ( select name from sysobjects where name = 'sp10tpcc_neworder' )
drop procedure sp10tpcc_neworder

go

create proc sp10tpcc_neworder

```

```

int,
tinyint,
int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare  @w_tax    numeric(4,4),
    @d_tax    numeric(4,4),
    @c_last   char(16),
    @c_credit char(2),
    @c_discount numeric(4,4),

    @i_price  numeric(5,2),
    @i_name   char(24),

```

```

@w_id    @o_entry_d char(50),
@d_id    @remote_flag int,
@c_id    @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 available 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 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 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 4 then
then s_dist_01
then s_dist_02
when 5 then
then s_dist_03
when 6 then
then s_dist_04
when 7 then
then s_dist_05
when 8 then
then s_dist_06
when 9 then
then s_dist_07
when 10 then
then s_dist_08
when 11 then
then s_dist_09
when 12 then
10 then s_dist_10
and
where       s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@l_price * @li_qty,

when 1
when 2
            @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,

```

```

@o_entry_d,
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 (repeatable)
where w_id = @w_id
if (@commit_flag = 1)
commit transaction n
else
rollback transaction n
-----
-- 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
if exists ( select name from sysobjects where name = 'sp11tpcc_neworder' )
drop procedure sp11tpcc_neworder
go

create proc sp11tpcc_neworder
int,
tinyint,
int,
@o_ol_cnt tinyint,
@o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
as
declare @w_tax numeric(4,4),
@d_tax numeric(4,4),
@c_last char(16),
@c_credit char(2),
@c_discount numeric(4,4),
@i_price numeric(5,2),
@i_name char(24),
@w_id
@d_id
@c_id
@o_entry_d 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 available 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,

when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end

then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id
where
s_i_id
s_w_id
end
= @li_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @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,
                                     @o_entry_d,
                                     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)
        else
        commit transaction n
-----
-- 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
        if exists ( select name from sysobjects where name = 'sp12tpcc_neworder' )
        drop procedure sp12tpcc_neworder
        go
        create proc sp12tpcc_neworder
        as
        declare      @w_tax      numeric(4,4),
                   @d_tax      numeric(4,4),
                   @c_last     char(16),
                   @c_credit   char(2),
                   @c_discount numeric(4,4),
                   @i_price    numeric(5,2),
                   @i_name     char(24),
                   @i_data     char(50),
                   @w_id
                   @d_id
                   @c_id
                   @o_ol_cnt tinyint,
                   @o_all_local tinyint,
                   @i_id1
                   @i_id2
                   @i_id3
                   @i_id4
                   @i_id5
                   @i_id6
                   @i_id7
                   @i_id8
                   @i_id9
                   @i_id10
                   @i_id11
                   @i_id12
                   @i_id13
                   @i_id14
                   @i_id15
                   int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
                   int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
                   int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
                   int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
                   int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
                   int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
                   int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
                   int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
                   int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
                   int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
                   int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
                   int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
                   int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
                   int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
                   int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
    
```

```

@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 available 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 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,

                @ol_qty = case @li_no
                when 1 then
                when 2 then
                when 3 then
                when 4 then
                when 5 then
                when 6 then
                when 7 then
                when 8 then
                when 9 then
                when 10 then
                when 11 then
                when 12 then
                when 13 then
                when 14 then
                when 15 then
                end,

                @li_s_w_id = case @li_no
                when 1 then
                when 2 then
                when 3 then
                when 4 then
                when 5 then
                when 6 then
                when 7 then
                when 8 then
                when 9 then
                when 10 then
                when 11 then
                when 12 then
                when 13 then
                when 14 then
                when 15 then
                end,

                @ol_qty4
                @ol_qty5
                @ol_qty6
                @ol_qty7
                @ol_qty8
                @ol_qty9
                @ol_qty10
                @ol_qty11
                @ol_qty12
                @ol_qty13
                @ol_qty14
                @ol_qty15
                end,

                @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

```



```

then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

where s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,

```

```

when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
end
end
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,
@o_entry_d,
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
if exists ( select name from sysobjects where name = 'sp13tpcc_neworder' )
drop procedure sp13tpcc_neworder
go

```

```

create proc sp13tpcc_neworder
int,
tinyint,
int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
as
declare @w_tax      numeric(4,4),
        @d_tax      numeric(4,4),
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount numeric(4,4),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

        @i_price     numeric(9,2),
        @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 available 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
        end
-----
-- 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 4 then
then s_dist_01
when 5 then
then s_dist_02
when 6 then
then s_dist_03
when 7 then
then s_dist_04
when 8 then
then s_dist_05
when 9 then
then s_dist_06
when 10 then
then s_dist_07
when 11 then
then s_dist_08
when 12 then
then s_dist_09
10 then s_dist_10
and
where       s_i_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@l_price * @li_qty,

when 1
when 2
            @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,

```

```

        @o_entry_d,
        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 (repeatable)
where       w_id        = @w_id

if (@commit_flag = 1)
    commit transaction n
else
    rollback transaction n

-----
-- 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
if exists ( select name from sysobjects where name = 'sp14tpcc_neworder' )
    drop procedure sp14tpcc_neworder
go

create proc sp14tpcc_neworder
int,
tinyint,
int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare      @w_tax      numeric(4,4),
            @d_tax      numeric(4,4),
            @c_last     char(16),
            @c_credit   char(2),
            @c_discount numeric(4,4),
            @i_price    numeric(5,2),
            @i_name     char(24),
            @w_id       int,
            @d_id       int,
            @c_id       int,
            @o_id       int,
            @o_ol_cnt   tinyint,
            @o_all_local tinyint,
            @i_id1      int,
            @i_id2      int,
            @i_id3      int,
            @i_id4      int,
            @i_id5      int,
            @i_id6      int,
            @i_id7      int,
            @i_id8      int,
            @i_id9      int,
            @i_id10     int,
            @i_id11     int,
            @i_id12     int,
            @i_id13     int,
            @i_id14     int,
            @i_id15     int,
            @o_entry_d  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 available 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
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
end,

@li_qty = case @li_no
when 1 then
@ol_qty1
@ol_qty2
@ol_qty3
when 2 then
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15
when 3 then
@ol_qty1
@ol_qty2
@ol_qty3
when 4 then
@ol_qty4
@ol_qty5
when 5 then
@ol_qty6
when 6 then
@ol_qty7
when 7 then
@ol_qty8
when 8 then
@ol_qty9
when 9 then
@ol_qty10
when 10 then
@ol_qty11
when 11 then
@ol_qty12
when 12 then
@ol_qty13
when 13 then
@ol_qty14
when 14 then
@ol_qty15
when 15 then
@ol_qty15
end
end

-----
-- get item data (no one updates item)
-----
select @i_price = i_price,
@i_name = i_name,
@i_data = i_data
from item WITH (repeatable read)
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 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end
then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id
where s_i_id = @li_id
s_w_id =
end
-----
-- 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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @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,
                                   @o_entry_d,
                                   @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
                rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
        if exists ( select name from sysobjects where name = 'sp15tpcc_neworder' )
                drop procedure sp15tpcc_neworder
        go
        create proc sp15tpcc_neworder
        as
        declare      @w_tax      numeric(4,4),
                   @d_tax      numeric(4,4),
                   @c_last     char(16),
                   @c_credit   char(2),
                   @c_discount numeric(4,4),
                   @i_price    numeric(5,2),
                   @i_name     char(24),
                   @i_data     char(50),
                   @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
    
```

```

@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 available 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 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,
                        @ol_qty = case @li_no
                        when 1 then @ol_qty4
                        when 2 then @ol_qty5
                        when 3 then @ol_qty6
                        when 4 then @ol_qty7
                        when 5 then @ol_qty8
                        when 6 then @ol_qty9
                        when 7 then @ol_qty10
                        when 8 then @ol_qty11
                        when 9 then @ol_qty12
                        when 10 then @ol_qty13
                        when 11 then @ol_qty14
                        when 12 then @ol_qty15
                        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 @li_qty1
                        when 2 then @li_qty2
                        when 3 then @li_qty3
                        when 4 then @li_qty4
                        when 5 then @li_qty5
                        when 6 then @li_qty6
                        when 7 then @li_qty7
                        when 8 then @li_qty8
                        when 9 then @li_qty9
                        when 10 then @li_qty10
                        when 11 then @li_qty11
                        when 12 then @li_qty12
                        when 13 then @li_qty13
                        when 14 then @li_qty14
                        when 15 then @li_qty15
                        end,
                        @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

```

```

then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

and             where      s_i_id
                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,
                @li_s_w_id,
                'dec 31, 1899',
                @li_qty,

when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
end
=

                @i_price * @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","0,0,0
                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,
                @o_entry_d,
                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

-----
if exists ( select name from sysobjects where name = 'sp16tpcc_neworder' )

```



```

drop procedure sp16tpcc_neworder
go
create proc sp16tpcc_neworder
int,
tinyint,
int,
        @o_ol_cnt    tinyint,
        @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
as
declare    @w_tax      numeric(4,4),
          @d_tax      numeric(4,4),
          @c_last     char(16),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@c_desc    char(24),
@i_price   numeric(5,2),
@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 available 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
        end
-----
-- set l_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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2

```

```

when 3 then @i_id3
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
              when 2 then
              when 3 then
              when 4 then
              when 5 then
              when 6 then
              when 7 then
              when 8 then
              when 9 then
              when 10 then
              when 11 then
              when 12 then
              when 13 then
              when 14 then
              when 15 then
              end,
@li_qty = case @li_no
            when 1 then
            when 2 then

```

```

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,

when 3 then
    @d_id
when 4 then
    then s_dist_01
when 5 then
    then s_dist_02
when 6 then
    then s_dist_03
when 7 then
    then s_dist_04
when 8 then
    then s_dist_05
when 9 then
    then s_dist_06
when 10 then
    then s_dist_07
when 11 then
    then s_dist_08
when 12 then
    then s_dist_09
when 13 then
    10 then s_dist_10
when 14 then
    and
where
    s_i_id
when 15 then
    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,
    @li_s_w_id,
    'dec 31, 1899',
    @li_qty,

= case
when 1
    @i_price * @li_qty,
@s_dist)
when 2
when 3
    -----
    -- 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_entry_d,
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 (repeatable read)
where w_id = @w_id

if (@commit_flag = 1)
    commit transaction n
else
    rollback transaction n
-----
-- 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

if exists ( select name from sysobjects where name = 'sp17tpcc_neworder' )

```

```

go
drop procedure sp17tpcc_neworder
create proc sp17tpcc_neworder
@w_id int,
@d_id tinyint,
@c_id int,
@o_ol_cnt tinyint,
@o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
@d_tax numeric(4,4),
@c_last char(16),
@c_credit char(2),

```

```

@w_id @i_price numeric(5,2),
@d_id @i_name char(24),
@c_id @i_data char(50),
@o_entry_d @o_entry_d datetime,
@remote_flag @remote_flag int,
@s_quantity @s_quantity smallint,
@s_data @s_data char(50),
@s_dist @s_dist char(24),
@li_no @li_no int,
@o_id @o_id int,
@commit_flag @commit_flag tinyint,
@li_id @li_id int,
@li_s_w_id @li_s_w_id int,
@li_qty @li_qty smallint,
@ol_number @ol_number int,
@c_id_local @c_id_local int

begin
begin transaction n
-----
-- get district tax and next available 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
@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,

when 2 then
when 3 then
@d_id
then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id
where
s_i_id
s_w_id
= s_data,
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_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,
@li_s_w_id,
'dec 31, 1899',

```

```

        @i_qty,
        @i_price * @i_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 * @i_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,

```

```

        @c_id_local,
        @o_entry_d,
        @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
                rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
if exists ( select name from sysobjects where name = 'sp18tpcc_neworder' )

```

```

        @w_id,
go        drop procedure sp18tpcc_neworder
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      numeric(4,4),
             @d_tax      numeric(4,4),
             @c_last     char(16),
             @c_credit   char(2),

```

```

@c_discount numeric(4,4),
@i_price    numeric(5,2),
@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 available 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,
           @ol_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,
           @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,
           @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              @li_qty,
                  @i_price * @li_qty,
then s_dist_01    @s_dist)
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10

        where      s_i_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,
        @li_s_w_id,
        'dec 31, 1899',

        @c_id_local,
        @o_entry_d,
        @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
                rollback transaction n

-----
-- 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
if exists ( select name from sysobjects where name = 'sp19tpcc_neworder' )
        drop procedure sp19tpcc_neworder

```

```

go
create proc sp19tpcc_neworder
int,
tinyint,
int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare    @w_tax    numeric(4,4),
    @d_tax    numeric(4,4),
    @c_last   char(16),
    @c_credit char(2),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

    @c_discount numeric(5,4),
    @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 available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2

```

```

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
                when 2 then
                when 3 then
                when 4 then
                when 5 then
                when 6 then
                when 7 then
                when 8 then
                when 9 then
                when 10 then
                when 11 then
                when 12 then
                when 13 then
                when 14 then
                when 15 then
                end,

@li_qty = case @li_no
                when 1 then
                when 2 then

```



```

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,

when 3 then
@d_id
when 4 then
then s_dist_01
when 5 then
then s_dist_02
when 6 then
then s_dist_03
when 7 then
then s_dist_04
when 8 then
then s_dist_05
when 9 then
then s_dist_06
when 10 then
then s_dist_07
when 11 then
then s_dist_08
when 12 then
then s_dist_09
when 13 then
10 then s_dist_10
and
where      s_i_id
            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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,

= case
when 1
@i_price * @li_qty,
@s_dist)
when 2
when 3
-----
-- 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,
@o_entry_d,
@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
    rollback transaction n
-----
-- 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

if exists ( select name from sysobjects where name = 'sp20tpcc_neworder' )
    drop procedure sp20tpcc_neworder

go
create proc sp20tpcc_neworder
int,
tinyint,
int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @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 available 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

```

```

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
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
end,
@li_qty = case @li_no
when 1 then
@ol_qty1
@ol_qty2
when 2 then
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15
end,
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
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,
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end
-----
-- 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,
                             @li_s_w_id,
                             'dec 31, 1899',
                             @li_qty,
                             @s_dist
                             = case
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=
)

```

```

        @i_price * @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
                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_entry_d,
        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
                rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
        if exists ( select name from sysobjects where name = 'sp21tpcc_neworder' )
                drop procedure sp21tpcc_neworder
        go

```

```

create proc sp21tpcc_neworder
int,
tinyint,
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 numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @i_price numeric(5,2),

```

```

    @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 available 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,
                @ol_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,
                @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,
    end
end

```

```

@s_dist
@d_id
then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id

where s_i_id
s_w_id

=

= case
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id

-- no item (or stock) found - triggers rollback condition

select "0","0","0,0,0
select @commit_flag = 0

end

-- get customer last name, discount, and credit rating

select
from
where
customer WITH (repeatableread)
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,

@c_id_local,
@o_entry_d,
@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
rollback transaction n

-- all that work for nuthin!!!

-- 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
if exists ( select name from sysobjects where name = 'sp22tpcc_neworder' )
drop procedure sp22tpcc_neworder

```

```

go
create proc sp22tpcc_neworder
int,
tinyint,
int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare  @w_tax      numeric(4,4),
         @d_tax      numeric(4,4),
         @c_last     char(16),
         @c_credit   char(2),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@c_discount numeric(5,4),
@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 available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2

```

```

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
when 2 then
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end,

@li_qty = case @li_no
when 1 then
when 2 then

```

```

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,

when 3 then
@d_id
then s_dist_01
when 4 then
then s_dist_02
when 5 then
then s_dist_03
when 6 then
then s_dist_04
when 7 then
then s_dist_05
when 8 then
then s_dist_06
when 9 then
then s_dist_07
when 10 then
then s_dist_08
when 11 then
then s_dist_09
when 12 then
10 then s_dist_10
and
where      s_i_id
           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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,

= case
when 1
@i_price * @li_qty,
@s_dist)
when 2
when 3
-----
-- 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,
@o_entry_d,
@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
    rollback transaction n
-----
-- 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

if exists ( select name from sysobjects where name = 'sp23tpcc_neworder' )
    drop procedure sp23tpcc_neworder

go
create proc sp23tpcc_neworder
int,
tinyint,
int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),
        @c_last char(16),
        @c_credit char(2),
        @c_discount numeric(4,4),
        @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 available 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

```

```

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
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
end,
@li_qty = case @li_no
when 1 then
@ol_qty1
@ol_qty2
when 2 then
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15
end,
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
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,
when 3 then
when 4 then
@d_id
then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id
where
s_i_id
s_w_id
= case
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,

```

```

        @i_price * @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_entry_d,
        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
                rollback transaction n
-----
-- 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

```

```

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 @w_tax numeric(4,4),
        @d_tax numeric(4,4),

```

```

@c_last      char(16),
@c_credit    char(2),
@c_discount  numeric(4,4),
@i_price     numeric(5,2),
@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 available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

```

```

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
              when 2 then
              when 3 then
              when 4 then
              when 5 then
              when 6 then
              when 7 then
              when 8 then
              when 9 then
              when 10 then
              when 11 then
              when 12 then
              when 13 then
              when 14 then
              when 15 then
              end,

@li_qty = case @li_no

```

```

@ol_qty1
@ol_qty2
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15

-----
-- 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 - @li_qty +
                   @s_quantity = s_quantity -
        case when (s_quantity - @li_qty < 10) then 91 else 0 end,
        s_order_cnt = s_order_cnt + 1,
end

```

```

s_remote_cnt + case when (@li_s_w_id = @w_id) then 0 else 1 end,
s_remote_cnt =
@s_data = s_data,
@s_dist = case
@d_id
then s_dist_01
then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@li_s_w_id
where s_i_id = @li_id
and 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,
@li_s_w_id,

```

```

'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)
-----
when 1
when 2
when 3 -- 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,
@d_id,
@c_id_local,
@o_entry_d,
0,
@o_ol_cnt,
@o_all_local)
-----
-- insert corresponding row into new-order table
-----
insert into new_order values ( @o_id,
@d_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
-----
-- all that work for nuthin!!!
-----
-- 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

```

```

if exists ( select name from sysobjects where name = 'sp25tpcc_neworder' )
    drop procedure sp25tpcc_neworder
go

create proc sp25tpcc_neworder
int,
tinyint,
int,
        @o_ol_cnt    tinyint,
        @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare    @w_tax    numeric(4,4),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@c_last    char(16),
@c_credit  char(2),
@c_discount numeric(4,4),
@i_price   numeric(5,2),
@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 available 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 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,

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

@s_w_id = case @li_no
        when 1 then
        when 2 then
        when 3 then
        when 4 then
        when 5 then
        when 6 then
        when 7 then
        when 8 then
        when 9 then
        when 10 then
        when 11 then
        when 12 then
        when 13 then
        when 14 then
        when 15 then
        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_remote_cnt = s_order_cnt + 1,
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,
@_li_s_w_id,

```

```

'dec 31, 1899',
@li_qty,
@_i_price * @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_entry_d,

                           @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 (repeatable read)
where w_id = @w_id

if (@commit_flag = 1)
    commit transaction n
else
    rollback transaction n

-----
-- 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

```

```

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,

    int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
        @d_tax numeric(4,4),

```

```

@c_credit char(26),
@c_discount numeric(4,4),
@i_price numeric(5,2),
@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 available 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,

when 1 then
when 2 then
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end

s_remote_cnt =
s_remote_cnt + case when (@li_s_w_id = @s_w_id) then 0 else 1 end,
@s_dist = case
when 1
when 2
when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=
and
@li_s_w_id
s_i_id
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,
@s_d_id,
@s_w_id,
@li_no,
@li_id,
@li_s_w_id,

```

```

'dec 31, 1899',

@li_qty,

@i_price * @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,

            @c_id_local,

            @o_entry_d,

            @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

```

```

@d_id,
@w_id,

0,

@d_id,
@w_id)

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 numeric(4,4),
            @d_tax numeric(4,4),

```

```

@c_last      char(16),
@c_credit    char(2),
@c_discount  numeric(4,4),
@i_price     numeric(5,2),
@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 available 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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15

```

```

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
              when 2 then
              when 3 then
              when 4 then
              when 5 then
              when 6 then
              when 7 then
              when 8 then
              when 9 then
              when 10 then
              when 11 then
              when 12 then
              when 13 then
              when 14 then
              when 15 then
              end,

@li_qty = case @li_no

```

```

when 1 then
when 2 then

when 3 then

when 4 then

when 5 then

when 6 then

when 7 then

when 8 then

when 9 then

when 10 then

when 11 then

when 12 then

when 13 then

when 14 then

when 15 then

end

-----
-- get item data (no one updates item)
-----
        select      @i_price = i_price,
                    @i_name  = i_name,
                    @i_data  = i_data
        from        item WITH (repeatable)
        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 + case when (@li_s_w_id = @w_id) then 0 else 1 end,
s_remote_cnt =
@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,
@li_s_w_id,

```

```

'dec 31, 1899',
@li_qty,
@i_price * @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,
@d_id,
@c_id_local,
@o_entry_d,
0,
@o_ol_cnt,
@o_all_local)
-----
-- insert corresponding row into new-order table
-----
insert into new_order values ( @o_id,
@d_id,
@d_id,
@d_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

```

```

if exists ( select name from sysobjects where name = 'sp28tpcc_neworder' )
    drop procedure sp28tpcc_neworder
go
create proc sp28tpcc_neworder
int,
tinyint,
int,
        @o_ol_cnt    tinyint,
        @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
as
declare @w_tax    numeric(4,4),
        @d_tax    numeric(4,4),
        @c_last   char(16),

```

```

@w_id
@d_id
@c_id
@i_id1
@i_id2
@i_id3
@i_id4
@i_id5
@i_id6
@i_id7
@i_id8
@i_id9
@i_id10
@i_id11
@i_id12
@i_id13
@i_id14
@i_id15

```

```

@c_desc    char(24),
@i_price    numeric(5,2),
@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 available 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
        end
-----
-- set l_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

```

```

@s_w_id1
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
@ol_qty1
@ol_qty2

```

```

when 3 then @i_id3
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
                    when 2 then
                    when 3 then
                    when 4 then
                    when 5 then
                    when 6 then
                    when 7 then
                    when 8 then
                    when 9 then
                    when 10 then
                    when 11 then
                    when 12 then
                    when 13 then
                    when 14 then
                    when 15 then
                    end,
        @li_qty = case @li_no
                    when 1 then
                    when 2 then

```

```

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
when 1
    @d_id
    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
= @li_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,
    @li_s_w_id,
    'dec 31, 1899',
    @li_qty,

```

```

@i_price * @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,

```

```

@c_id_local,

@o_entry_d,

@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 (repeatable)
where w_id = @w_id

if (@commit_flag = 1)
    commit transaction n
else
    rollback transaction n

-----
-- all that work for nuthin!!!
-----

-- 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
if exists ( select name from sysobjects where name = 'sp29tpcc_neworder' )
    drop procedure sp29tpcc_neworder

```

```

go
create proc sp29tpcc_neworder

int,

tinyint,

int,

    @o_ol_cnt tinyint,

    @o_all_local tinyint,

int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare @w_tax numeric(4,4),
@d_tax numeric(4,4),
@c_last char(16),
@c_credit char(2),
@c_discount numeric(4,4),

```

```

@w_id

@d_id

@c_id

@i_id1

@i_id2

@i_id3

@i_id4

@i_id5

@i_id6

@i_id7

@i_id8

@i_id9

@i_id10

@i_id11

@i_id12

@i_id13

@i_id14

@i_id15

@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 available 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 @i_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
@s_w_id2
@s_w_id3
@s_w_id4
@s_w_id5
@s_w_id6
@s_w_id7
@s_w_id8
@s_w_id9
@s_w_id10
@s_w_id11
@s_w_id12
@s_w_id13
@s_w_id14
@s_w_id15
when 2 then
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end,
@li_qty = case @li_no
when 1 then
@ol_qty1
@ol_qty2
when 2 then
@ol_qty3
@ol_qty4
@ol_qty5
@ol_qty6
@ol_qty7
@ol_qty8
@ol_qty9
@ol_qty10
@ol_qty11
@ol_qty12
@ol_qty13
@ol_qty14
@ol_qty15
when 3 then
when 4 then
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
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
when 3 then
when 4 then
then s_dist_01
then s_dist_02
when 5 then
when 6 then
when 7 then
when 8 then
when 9 then
when 10 then
when 11 then
when 12 then
when 13 then
when 14 then
when 15 then
end
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @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,
                                   @o_entry_d,
                                   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)
        else
                commit transaction n
        else
                rollback transaction n
-----
-- all that work for nuthin!!!
-----
-- 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
if exists ( select name from sysobjects where name = 'sp30tpcc_neworder' )
drop procedure sp30tpcc_neworder
go
create proc sp30tpcc_neworder
        as
declare      @w_tax      numeric(4,4),
            @d_tax      numeric(4,4),
            @c_last     char(16),
            @c_credit   char(2),
            @c_discount numeric(4,4),
            @i_price    numeric(5,2),
            @i_name     char(24),
            @i_data     char(50),
            @w_id
            @d_id
            @c_id
            @o_ol_cnt   tinyint,
            @o_all_local tinyint,
            @i_id1
            @i_id2
            @i_id3
            @i_id4
            @i_id5
            @i_id6
            @i_id7
            @i_id8
            @i_id9
            @i_id10
            @i_id11
            @i_id12
            @i_id13
            @i_id14
            @i_id15
            int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
            int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
            int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
            int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
            int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
            int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
            int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
            int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
            int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
            int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
            int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
            int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
            int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
            int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
            int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

```

```

@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 available 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,
                        @ol_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,
                        @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 @li_qty1
                                    when 2 then @li_qty2
                                    when 3 then @li_qty3
                                    when 4 then @li_qty4
                                    when 5 then @li_qty5
                                    when 6 then @li_qty6
                                    when 7 then @li_qty7
                                    when 8 then @li_qty8
                                    when 9 then @li_qty9
                                    when 10 then @li_qty10
                                    when 11 then @li_qty11
                                    when 12 then @li_qty12
                                    when 13 then @li_qty13
                                    when 14 then @li_qty14
                                    when 15 then @li_qty15
                                    end
                        end

              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 - @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
                        when 1
then s_dist_01

```

```

then s_dist_02
then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and
@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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)

when 2 -----
-- send line-item data to client
when 3 -----
select @i_name,
@s_quantity,
b_g = case when (
when 4 (patindex('%ORIGINAL%',@i_data) > 0) and
when 5 (patindex('%ORIGINAL%',@s_data) > 0) )
then 'B' else 'G'
when 6 end,
when 7 @i_price,
when 8 @i_price * @li_qty
when 9 end
else
when 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,
@o_entry_d,
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
if exists ( select name from sysobjects where name = 'sp31tpcc_neworder' )
drop procedure sp31tpcc_neworder
go
create proc sp31tpcc_neworder
int,
@w_id

```

```

tinyint,
int,
    @o_ol_cnt    tinyint,
    @o_all_local tinyint,
int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

as
declare    @w_tax    numeric(4,4),
    @d_tax    numeric(4,4),
    @c_last   char(16),
    @c_credit char(2),
    @c_discount numeric(4,4),
    @i_price  numeric(5,2),
    @i_name   char(24),
    @i_data   char(50),
    @o_entry_d datetime,

    @d_id    int,
    @c_id    int,
    @i_id1   int,
    @i_id2   int,
    @i_id3   int,
    @i_id4   int,
    @i_id5   int,
    @i_id6   int,
    @i_id7   int,
    @i_id8   int,
    @i_id9   int,
    @i_id10  int,
    @i_id11  int,
    @i_id12  int,
    @i_id13  int,
    @i_id14  int,
    @i_id15  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 available 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
                    when 2 then
                    when 3 then
                    when 4 then
                    when 5 then
                    when 6 then
                    when 7 then
                    when 8 then
                    when 9 then
                    when 10 then
                    when 11 then
                    when 12 then
                    when 13 then
                    when 14 then
                    when 15 then
                end,

                @li_qty = case @li_no
                    when 1 then
                    when 2 then
                    when 3 then
                    when 4 then
                    when 5 then
                end,

                @ol_qty1
                @ol_qty2
                @ol_qty3
                @ol_qty4
                @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

```

```

then s_dist_03
then s_dist_04
then s_dist_05
then s_dist_06
then s_dist_07
then s_dist_08
then s_dist_09
10 then s_dist_10
and          where    s_i_id
              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,
@li_s_w_id,
'dec 31, 1899',
@li_qty,
@i_price * @li_qty,
@s_dist)

```

```

when 3
when 4
when 5
when 6
when 7
when 8
when 9
when
end
= @li_id
=

-----
-- 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_entry_d,
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
    rollback transaction n

-----
-- 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\_32-ordstatproc.sql**

```

-----
-- File: ORDSTAT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
-----
-- - Creates order status transaction stored procedure
-- - Interface Level: 4.10.000
-----

use tpcc
go

if exists ( select name from sysobjects where name = 'tpcc_orderstatus' )
    drop procedure tpcc_orderstatus
go

create proc tpcc_orderstatus          @w_id      int,
                                     @d_id      tinyint,
                                     @c_id      int,
                                     @c_last    char(16)
as
declare @c_balance                    numeric(12,2),
        @c_first                      char(16),
        @c_middle                    char(2),
        @o_id                        int,
        @o_entry_d                   datetime,
        @o_carrier_id                smallint,
        @cnt                          smallint

begin tran 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
    end

```

```

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 (repeatable)
where       ol_o_id = @o_id and
            ol_d_id = @d_id and
            ol_w_id = @w_id

custnotfound:

commit tran 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

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 numeric(12,2),
        @c_first   char(16),
        @c_middle  char(2),
        @o_id      int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt       smallint

begin tran o

if (@c_id = 0)
begin
-----
-- get customer id and info using last name
-----
select      @cnt = (count(*)+1)/2
from        customer WITH (repeatable)
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 (repeatable)
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 (repeatable)
where       c_id = @c_id

and

and

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 (repeatable)
where       ol_o_id = @o_id and
            ol_d_id = @d_id and
            ol_w_id = @w_id

```

```

custnotfound:

commit tran 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

if exists ( select name from sysobjects where name = 'sp1tpcc_orderstatus' )
    drop procedure  sp1tpcc_orderstatus
go

create proc sp1tpcc_orderstatus      @w_id      int,
                                    @d_id      tinyint,

                                    @c_id      int,
                                    @c_last   char(16)

= "

as

declare @c_balance      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran o

if (@c_id = 0)
    begin

-----
-- get customer id and info using last name
-----

        select      @cnt      = (count(*)+1)/2
        from        customer WITH (repeatableread)

```

```

where      @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
and
           c_w_id     = @w_id

           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 tran 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

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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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

```

```

        -----
        -- return data to client
        -----
        select @c_id,
               @c_last,
               @c_first,
               @c_middle,
               @o_entry_d,
               @o_carrier_id,
               @c_balance,
               @o_id

        go
        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 numeric(12,2),
                @c_first char(16),
                @c_middle char(2),
                @o_id int,
                @o_entry_d datetime,
                @o_carrier_id smallint,
                @cnt smallint

        begin tran 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

```

```

-----
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
and
            c_w_id    = @w_id
            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 tran 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
if exists ( select name from sysobjects where name = 'sp4tpcc_orderstatus' )
drop procedure sp4tpcc_orderstatus
go

create proc sp4tpcc_orderstatus @w_id tinyint,
                                @c_id int,
                                @c_last char(16)
as
declare @c_balance numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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
and
            c_w_id    = @w_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

```

```

where       ol_d_id = @d_id and
            ol_w_id = @w_id

custnotfound:

commit tran 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
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      numeric(12,2),
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

begin tran 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_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 tran 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
if exists ( select name from sysobjects where name = 'sp6tpcc_orderstatus' )
    drop procedure sp6tpcc_orderstatus
go

```

```

create proc sp6tpcc_orderstatus @w_id tinyint,
                                @d_id tinyint,
                                @c_id int,
                                @c_last char(16)

= "

as

declare @c_balance numeric(12,2),
        @c_first   char(16),
        @c_middle  char(2),
        @o_id      int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt       smallint

begin tran 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

select      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

select      c_d_id = @d_id

and

select      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 tran 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
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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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
and
      c_w_id = @w_id

      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 tran 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

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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
begin tran 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_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 tran 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
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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
begin tran 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
and
            c_w_id    = @w_id

            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 tran 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

```

```

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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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
and
            c_w_id    = @w_id

            c_d_id    = @d_id
order       by c_w_id, c_d_id, c_last, c_first

```

```

end      set      rowcount 0
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    @l_supply_w_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 tran 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
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    numeric(12,2),
        @c_first      char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @cnt          smallint

begin tran 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_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 tran 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
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      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran 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

```

```

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 tran 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
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      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

```

```

begin tran 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

and

c_w_id      = @w_id

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

```

```

end
select      @cnt      = @@rowcount

-----
-- 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 tran 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
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    numeric(12,2),
        @c_first      char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @cnt          smallint
begin tran 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
        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 tran 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
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    numeric(12,2),
        @c_first      char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,

```

```

        @o_carrier_id      smallint,
        @cnt              smallint

begin tran 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
and
        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 tran o

        -----
        -- return data to client
        -----
        select      @c_id,
                   @c_last,
                   @c_first,

```

```

        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

go

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 numeric(12,2),
        @c_first   char(16),
        @c_middle  char(2),
        @o_id      int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt       smallint

begin tran 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_d_id = @d_id and
    order o_w_id = @w_id
    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 tran 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
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 char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
begin tran 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)

```

```

and          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 tran 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
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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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_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 (repeatable read)
where ol_o_id = @o_id and
      ol_d_id = @d_id and
      ol_w_id = @w_id

custnotfound:

commit tran 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
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 numeric(12,2),
        @c_first   char(16),
        @c_middle  char(2),
        @o_id      int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt       smallint

begin tran o

if (@c_id = 0)
    begin
        -----
        -- get customer id and info using last name
        -----
        select @cnt = (count(*)+1)/2
        from customer WITH (repeatable read)
        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 (repeatable read)
        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,

```

```

from customer WITH (repeatable read)
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 (repeatable read)
where ol_o_id = @o_id and
      ol_d_id = @d_id and
      ol_w_id = @w_id

custnotfound:

commit tran 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
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      numeric(12,2),
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

begin tran 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

        and
        and

        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_entry_d = o_entry_d, @o_id,
                   @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 tran 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
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      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran 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

        set rowcount 0

        order by c_w_id, c_d_id, c_last, c_first

        end

        else
            begin
                -----
                -- get customer info if by id
                -----
                select @c_balance = c_balance,

```

```

        @e_middle = e_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 tran 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
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      numeric(12,2),
        @c_first       char(16),
        @c_middle      char(2),
        @o_id          int,
        @o_entry_d     datetime,
        @o_carrier_id  smallint,
        @cnt           smallint

begin tran 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
and
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
and
c_d_id = @d_id
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 tran 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
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 numeric(12,2),
@c_first char(16),
@c_middle char(2),
@o_id int,
@o_entry_d datetime,
@o_carrier_id smallint,
@cnt smallint
begin tran 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
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 (repeatable read)
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 (repeatable read)
where       ol_o_id = @o_id and
            ol_d_id = @d_id and
            ol_w_id = @w_id

```

```

custnotfound:
commit tran 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

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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran o

if (@c_id = 0)
begin

-----
-- get customer id and info using last name
-----

```

```

select      customer WITH (repeatable read)
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 (repeatable read)
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 (repeatable read)
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 tran 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

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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

begin tran 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,

```

```

        from      ol_delivery_d
        where     order_line WITH (repeatableread)
                ol_o_id = @o_id and
                ol_d_id = @d_id and
                ol_w_id = @w_id

custnotfound:

commit tran 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
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      numeric(12,2),
        @c_first       char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @cnt         smallint

begin tran 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 tran 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
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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
begin tran 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_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 tran 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
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 numeric(12,2),
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint
begin tran 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
and
            c_w_id    = @w_id

            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 tran 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

```

```

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      numeric(12,2),
        @c_first        char(16),
        @c_middle       char(2),
        @o_id           int,
        @o_entry_d      datetime,
        @o_carrier_id   smallint,
        @cnt            smallint

begin tran 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
and
            c_w_id    = @w_id

            c_d_id    = @d_id
order       by c_w_id, c_d_id, c_last, c_first

```

```

end      set      rowcount 0
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    @l_supply_w_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 tran 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
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    numeric(12,2),
        @c_first     char(16),
        @c_middle    char(2),
        @o_id        int,
        @o_entry_d   datetime,
        @o_carrier_id smallint,
        @cnt         smallint

begin tran 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_w_id     = @w_id
          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 tran 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
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    numeric(12,2),
        @c_first      char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @cnt          smallint

begin tran 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

```

```

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 tran 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\_32-deliveryproc.sql

```

-----
-- File: DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - Creates delivery transaction stored procedure
--
-- - Interface Level: 4.10.000
-----

```

use tpcc  
go

```

if exists (select name from sysobjects where name = 'tpcc_delivery' )
drop procedure tpcc_delivery

```

```

create proc tpcc_delivery @w_id int,

```

```

@o_carrier_id

```

```

smallint
as

```

```

declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @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 tran 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,
                             update lock)
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 = @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,
    = 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 + ol_amount
where ol_w_id
    = @w_id and
    ol_d_id
    = @d_id and
    ol_o_id
    = @o_id

```

-- accumulate lineitem amounts for this order into customer

```

-----
update customer
set c_balance =
c_balance + @total,
    = 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 tran d

-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10

go

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 numeric(12,2),
@oid1 int,
@oid2 int,

```

```

@oid4 int,
@oid5 int,
@oid6 int,
@oid7 int,
@oid8 int,
@oid9 int,
@oid10int

select @d_id = 0

begin tran 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

where o_w_id
o_d_id

= @w_id and
@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

where ol_w_id
ol_d_id
ol_o_id

= @w_id and
@d_id and
@o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
c_delivery_cnt

where c_w_id
c_d_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 tran d

-----

```

```

-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

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 numeric(12,2),
        @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 tran 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,
                               update lock)

        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

            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

            where o_w_id

            and    o_d_id

            and    o_id

            -----
            -- set date in all lineitems for this order (and sum amounts)
            -----
            update order_line
            set    ol_delivery_d =
                   getdate(),
                   @total

            where  ol_w_id

            and    ol_d_id

            and    ol_o_id

            and    @o_id

```

```

----- accumulate 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 tran d

-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

if exists (select name from sysobjects where name = 'sp2tpcc_delivery' )
    drop procedure sp2tpcc_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 numeric(12,2),
        @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 tran 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,
        uplock)
        where no_w_id = @w_id and
              no_d_id = @d_id
        order by no_o_id asc
        if (@@rowcount <> 0)
        begin
            delete new_order
            where no_w_id = @w_id
            and
            no_d_id = @d_id
        end
    end

```

```

no_o_id = @o_id
-----
-- set carrier_id on this order (and get customer id)
-----
update orders
set o_carrier_id =
@o_carrier_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 + ol_amount
    where ol_w_id
    = @w_id and
    ol_d_id
    = @d_id and
    ol_o_id
    = @o_id
-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
    = 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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
go
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 numeric(12,2),
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

```

```

@oid10int
select @d_id = 0
begin tran 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),
        uplock)
        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
            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
-----
-- accumulate 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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,

```

```

@oid4,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10

go
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
                             int,
smallint
as

declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10int

select @d_id = 0
begin tran 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),
        uplock)
        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

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,

```

```

= c_delivery_cnt + 1 where c_delivery_cnt
= @w_id and c_w_id

= @d_id and c_d_id

= @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 tran d

-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10

go
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 numeric(12,2),
@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 tran 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,
    = o_c_id
    = @w_id and
    = @d_id and
    = @o_id

-----
-- set date in all lineitems for this order (and sum amounts)
-----
update order_line
set ol_delivery_d=
getdate(),
    = @total + ol_amount
    = @w_id and
    = @d_id and
    = @o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
    = c_delivery_cnt + 1
    = @w_id and
    = @d_id and
    = @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,
@oid10 = case @d_id when 10 then @o_id else @oid10 end
end
commit tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
@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 tran 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),
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,
    = o_c_id
    = @w_id and
    = @d_id and
    = @o_id
-----
-- set date in all lineitems for this order (and sum amounts)
-----
update order_line
set ol_delivery_d=
getdate(),
    = @total + ol_amount

```



```

        = @w_id and      where    ol_w_id
                                ol_d_id
        = @d_id and      ol_o_id
        = @o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
        update      customer
        set         c_balance =
c_balance + @total,
                                c_delivery_cnt
                                = c_delivery_cnt + 1
                                where
        = @w_id and      c_w_id
        = @d_id and      c_d_id
        = @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 tran d

-----
-- return delivery data to client
-----
select @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,

```

```

@oid9
go
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 int,
smallint
as

declare @d_id      tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @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 tran 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),

                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
        = @w_id and      o_w_id
        = @d_id and      o_d_id
        = @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
        = @w_id and      ol_w_id
        = @d_id and      ol_d_id
        = @o_id           ol_o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
        update      customer
        set         c_balance =
c_balance + @total,
                                c_delivery_cnt
                                = c_delivery_cnt + 1
                                where
        = @w_id and      c_w_id
        = @d_id and      c_d_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 tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 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 tran d
while (@d_id < 10)
begin
select @d_id = @d_id + 1,
@total = 0,
@o_id = 0
uplock)
select top 1
@o_id = no_o_id
from new_order WITH (serializable,
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,
= 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
-----
-- accumulate 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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
go

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 numeric(12,2),
        @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 tran 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,

updatelock)

    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

            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

            where ol_w_id

              = @w_id and

```

```

              = @d_id and
              ol_d_id

              = @o_id
              ol_o_id

            -----
            -- accumulate lineitem amounts for this order into customer
            -----

            update customer
            set c_balance =

c_balance + @total,
              update
              set 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

end,

select @oid1 = case @d_id when 1 then @o_id else @oid1

@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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

```

```

go
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 numeric(12,2),
        @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 tran 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),

        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
-----

```

```

-----
where new_order = @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,

        = o_c_id
        = @w_id and
        = @d_id and
        = @o_id

-----
-- set date in all lineitems for this order (and sum amounts)
-----
update order_line
set    ol_delivery_d=

getdate(),

        = @total + ol_amount
        = @w_id and
        = @d_id and
        = @o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set    c_balance =

c_balance + @total,

        = c_delivery_cnt + 1
        = @w_id and
        = @d_id and
        = @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 tran d

-----
-- return delivery data to client
-----
select @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10

go
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 numeric(12,2),
        @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 tran 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

= @w_id and

= @d_id and

= @o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set    c_balance =
c_balance + @total,
       c_delivery_cnt

= c_delivery_cnt + 1

= @w_id and

= @d_id and

= @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 tran d
-----

```

```

-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
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 numeric(12,2),
        @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 tran d

    while (@d_id < 10)
    begin

        select @d_id = @d_id + 1,
               @total = 0,
               @o_id = 0

        select top 1

```

```

update
  from      @o_id = no_o_id
            new_order WITH (serializable),

  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

            where  o_w_id

            = @w_id and

            = @d_id and

            = @o_id

-----
-- set date in all lineitems for this order (and sum amounts)
-----

            update order_line
            set    ol_delivery_d=

getdate(),

            @total

            where  ol_w_id

            = @w_id and

            = @d_id and

            = @o_id

-----
-- accumulate lineitem amounts for this order into customer

```

```

-----update-----
set customer
c_balance =

c_delivery_cnt

where c_w_id

c_d_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 tran d

-----
-- return delivery data to client
-----

select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10

go
if exists (select name from sysobjects where name = 'sp13tpcc_delivery')
drop procedure sp13tpcc_delivery
go

create proc sp13tpcc_delivery @w_id int,

```

```

smallint @o_carrier_id

as

declare @d_id tinyint,
@o_id int,
@c_id int,
@total numeric(12,2),
@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 tran 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),

uplock)
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,
      = o_c_id
      = @w_id and
      = @d_id and
      = @o_id
      where o_w_id
            o_d_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
      = @w_id and
      = @d_id and
      = @o_id
      where ol_w_id
            ol_d_id
            ol_o_id

-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
      c_delivery_cnt
      = c_delivery_cnt + 1
      = @w_id and
      = @d_id and
      = @c_id
      where c_w_id
            c_d_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,
      @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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
go
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 numeric(12,2),
       @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 tran 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,
                  = o_c_id
                  = @w_id and
                  = @d_id and
                  = @o_id
            -----
            -- set date in all lineitems for this order (and sum amounts)
            -----
            update order_line

```

```

getdate(),
set ol_delivery_d=
      @total
      = @total + ol_amount
      where ol_w_id
      = @w_id and
      ol_d_id
      = @d_id and
      ol_o_id
      = @o_id
-----
-- accumulate 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 tran d
-----
-- return delivery data to client
-----
select @oid1,
      @oid2,
      @oid3,
      @oid4,
      @oid5,

```

```

@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
        @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 tran 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,
            where no_w_id = @w_id and
                  no_d_id = @d_id
            order by no_o_id asc
uplock)

```

```

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
-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
      c_delivery_cnt
      = c_delivery_cnt + 1

```



```

        = @w_id and          where    c_w_id
                                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 tran d
-----
-- return delivery data to client
-----
select @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10
go
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 int,          tinyint,
        @c_id int,
        @total numeric(12,2),
        @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 tran 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
                                -----
                                -- accumulate lineitem amounts for this order into customer
                                -----
                                update    customer
                                set      c_balance =
                                c_balance + @total,
                                = 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 tran d

-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

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 numeric(12,2),
        @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 tran 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,

updatelock)
           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
        where  no_d_id = @d_id

and
        where  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

        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

```

```

           = @w_id and
           = @d_id and
           = @o_id

           where ol_w_id

           ol_d_id

           ol_o_id

-----
-- accumulate 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 tran d

-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,

```

```

@oid10
go
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 numeric(12,2),
        @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 tran 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)
        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
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
where o_w_id
      o_d_id
      o_id
      = o_c_id
      = @w_id and
      = @d_id and
      = @o_id
-----
-- set date in all lineitems for this order (and sum amounts)
-----
update order_line
set ol_delivery_d =
getdate(),
@total
where ol_w_id
      ol_d_id
      ol_o_id
      = @total + ol_amount
      = @w_id and
      = @d_id and
      = @o_id
-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
c_delivery_cnt
where c_w_id
      c_d_id
      = c_delivery_cnt + 1
      = @w_id and
      = @d_id and

```

```

      = @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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10
go
if exists (select name from sysobjects where name = 'sp19tpcc_delivery')
    drop procedure sp19tpcc_delivery
go
create proc sp19tpcc_delivery @w_id int,
                             @o_carrier_id
smallint
as
declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),

```

```

@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 tran 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),
        uplock)
        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
            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
        end
    end

```

```

    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
    where ol_w_id
           = @w_id and
           ol_d_id
           = @d_id and
           ol_o_id
           = @o_id

    -----
    -- accumulate lineitem amounts for this order into customer
    -----
    update customer
    set   c_balance =
    c_balance + @total,
        c_delivery_cnt
    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 tran d
-----
-- return delivery data to client
-----
select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
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 numeric(12,2),
        @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 tran 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)
    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
        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
        where ol_w_id = @w_id and
            ol_d_id = @d_id and
            ol_o_id = @o_id
    
```

```

        -----
        -- accumulate 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 tran d

    -----
    -- return delivery data to client
    -----
    select @oid1,
        @oid2,
        @oid3,
        @oid4,
        @oid5,
        @oid6,
        @oid7,
        @oid8,
        @oid9,
        @oid10

    go
    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 int
as
    declare @d_id tinyint,
        @o_id int,
        @c_id int,
        @total numeric(12,2),
        @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 tran 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)
            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
            end
        end
    
```

```

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
-----
-- accumulate 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,
@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 tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
@oid1 int,
@oid2 int,
@oid3 int,
@oid4 int,
@oid5 int,
@oid6 int,
@oid7 int,
@oid8 int,
@oid9 int,

```

```

@oid10int
select @d_id = 0
begin tran 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),
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
      = @w_id and
      = @d_id and
      = @o_id
-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,
      c_delivery_cnt
      = c_delivery_cnt + 1
      = @w_id and
      = @d_id and
      = @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 tran d
-----
-- return delivery data to client
-----
select @oid1,
      @oid2,
      @oid3,

```

```

@oid4,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
        @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 tran 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
      = @w_id and
      = @d_id and
      = @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
      = @w_id and
      = @d_id and
      = @o_id
-----
-- accumulate lineitem amounts for this order into customer
-----
update customer
set c_balance =
c_balance + @total,

```

```

        = c_delivery_cnt + 1
        = @w_id and
        = @d_id and
        = @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 tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
if exists (select name from sysobjects where name = 'sp24tpcc_delivery')
drop procedure sp24tpcc_delivery
go
create proc sp24tpcc_delivery @w_id int,
@c_id int,
@o_carrier_id
smallint

```

```

as
declare @d_id tinyint,
@o_id int,
@c_id int,
@total numeric(12,2),
@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 tran 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),
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,
= 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=
@total
getdate(),
@total
= @total + ol_amount
where ol_w_id
= @w_id and
ol_d_id
= @d_id and
ol_o_id
= @o_id
-----
-- accumulate 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 tran d

-----
-- return delivery data to client
-----

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go
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 numeric(12,2),
        @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 tran 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,
                           update lock)
    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,
               = 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 + ol_amount
    end
end

```

```

           where ol_w_id
                 ol_d_id
                 ol_o_id
           = @w_id and
           = @d_id and
           = @o_id

-----
-- accumulate 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
             c_d_id
             c_id
           = @w_id and
           = @d_id and
           = @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 tran d

-----
-- return delivery data to client
-----

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,

```

```

@oid10
go
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 numeric(12,2),
        @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 tran 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
= @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
-----
-- accumulate 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 tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
        @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 tran 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,
update)
        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

            -----
            -- accumulate 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 tran d

```

```

-----
-- return delivery data to client
-----

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

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 numeric(12,2),
        @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 tran 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),
update
  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
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
where ol_w_id =
  @w_id and
  ol_d_id =
  @d_id and
  ol_o_id =
  @o_id

```

```

-----
-- accumulate 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 tran d

-----
-- return delivery data to client
-----
select @oid1,
  @oid2,
  @oid3,
  @oid4,
  @oid5,
  @oid6,
  @oid7,
  @oid8,
  @oid9,
  @oid10
go
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 int,
smallint
as
declare @d_id tinyint,
  @o_id int,
  @c_id int,
  @total numeric(12,2),
  @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 tran 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),
    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
-----
-- accumulate 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,
@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 tran d
-----
-- return delivery data to client
-----
select @oid1,
@oid2,
@oid3,
@oid4,
@oid5,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
@oid1 int,
@oid2 int,
@oid3 int,
@oid4 int,
@oid5 int,
@oid6 int,
@oid7 int,
@oid8 int,
@oid9 int,

```

```

@oid10int
select @d_id = 0
begin tran 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(),
-- accummulate lineitem amounts for this order into customer
update customer
set c_balance =
c_balance + @total,
c_delivery_cnt
= c_delivery_cnt + 1
where ol_w_id
ol_d_id
ol_o_id
ol_o_id
= @total
where ol_w_id
ol_d_id
ol_o_id
= @w_id and
= @d_id and
= @o_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 tran d
-- return delivery data to client
select @oid1,
@oid2,
@oid3,

```

```

@oid4,
@oid6,
@oid7,
@oid8,
@oid9,
@oid10
go
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 numeric(12,2),
@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 tran 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,
where no_w_id = @w_id and
no_d_id = @d_id
order by no_o_id asc
updlock)

```

```

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
ol_d_id
ol_o_id
= @w_id and
= @d_id and
= @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 tran d

-----
-- return delivery data to client
-----

select @oid1,
       @oid2,
       @oid3,
       @oid4,
       @oid5,
       @oid6,
       @oid7,
       @oid8,
       @oid9,
       @oid10

go

-----
-- File: PAYMENT.SQL
-----

```

**createpaymentproc.sql**

```

Copyright Microsoft 2003
-- --
-- - Creates payment transaction stored procedure --
-- - Interface Level: 4.10.000 --
-----

use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_payment' )
drop procedure tpcc_payment
go

create proc tpcc_payment @w_id int,
                        @c_w_id int,
                        @h_amount
                        numeric(6,2),
                        @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 numeric(12,2),

```

```

@c_discount numeric(12,2),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = "

begin tran 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 -
      @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,
      @data = c_data,
      @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) +
458) substring(@data, 1,

```

```

-----update customer info
-----
update customer
set   c_data = @c_data
where c_id = @c_id and
      c_w_id = @c_w_id and
      c_d_id = @c_d_id
select @screen_data = substring
(@c_data,1,200)
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 tran 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
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 numeric(6,2),
                           @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

begin tran 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_h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@c_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,

```

```

        @data = c_data,
        @c_id_local = c_id = @c_id and
        c_id = @c_w_id and
        c_w_id = @c_d_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) +
                                substring(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
       c_w_id = @c_w_id and
       c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update district
set d_ytd = d_ytd +
@c_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 tran p

-----
-- return data to client
-----
select
    @c_id,
    @c_last,
    @datetime,
    @w_street_1,
    @w_street_2,
    @w_city,

```

```

    @w_state,
    @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
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 numeric(6,2),
    @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_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 numeric(12,2),
    @c_balance numeric(12,2),
    @c_discount numeric(4,4),
    @data char(500),
    @c_data char(500),
    @datetime datetime,
    @w_ytd numeric(12,2),
    @d_ytd numeric(12,2),
    @cnt smallint,
    @val smallint,
    @screen_data char(200),
    @d_id_local tinyint,
    @w_id_local int,
    @c_id_local int

select @screen_data = "

begin tran 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,
                  @data         = c_data,

        where     @c_id_local = c_id
                  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) +
        substring(@data, 1,
        458)

-----
-- update customer info
-----
        update    customer
        set       c_data = @c_data

        where     c_id = @c_id and
                  c_w_id = @c_w_id and
                  c_d_id = @c_d_id

        select    @screen_data = substring
        (@c_data,1,200)
        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 tran 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

```

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 numeric(6,2),
    @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),

```

```

        @w_id_local int,
        @c_id_local int
select @screen_data = ""
begin tran 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_h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@c_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_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,
        @data = c_data,
        @c_id_local = c_id,
        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) +
substr(@data, 1,
458)
-----
-- update customer info
-----
update customer
set c_data = @c_data
where c_id = @c_id and
       c_w_id = @c_w_id and
       c_d_id = @c_d_id
select @screen_data = substring
(@c_data,1,200)
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 tran 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
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   numeric(6,2),
                               @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_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 numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,
        @val        smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = ""

begin tran 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 -
                   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,
                   @data         = c_data,
                   @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) +
        substring(@data, 1,
        458)

-----
-- update customer info
-----
        update      customer
        set         c_data      = @c_data

        where       c_id      = @c_id and
                   c_w_id      = @c_w_id and
                   c_d_id      = @c_d_id

        select      @screen_data = substring
        (@c_data,1,200)
        end

-----
-- get district data and update year-to-date
-----
        update      district
        set         d_ytd      = d_ytd +
                   @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 tran 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
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     numeric(6,2),

    @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 numeric(12,2),
        @c_balance  numeric(12,2),

```

```

        @data       char(500),
        @c_count    char(4,4),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd     numeric(12,2),
        @d_ytd     numeric(12,2),
        @cnt       smallint,
        @val       smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

begin tran 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,
        @h_amount, c_ytd_payment = c_ytd_payment +

        @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,
        @data = c_data,

        @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) +

                substring(@data, 1,
                458)

-----
-- update customer info
-----

```

```

update customer
set c_data = @c_data

where c_id = @c_id and
      c_w_id = @c_w_id and
      c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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 tran 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
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 int,
                           @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

begin tran 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,

```

```

@data = c_data,
@c_id_local = 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) +
substring(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)

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_state = d_state,
@d_zip = d_zip,
@d_name = d_name,
@d_id_local = d_id = @d_id and
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 = @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 tran 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
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 numeric(6,2),
    @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_middle char(6),
@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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = ""

begin tran 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_d_id = @c_d_id and
order by c_last, c_first

set rowcount 0

end

-----
-- get customer info and update balances
-----
update customer
set @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,
@data = c_data,
@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) +
                                substring(@data, 1,
458)
-----
-- update customer info
-----
update      customer
set         c_data      = @c_data

where      c_id        = @c_id and
          c_w_id       = @c_w_id and
          c_d_id       = @c_d_id

select     @screen_data = substring
(@c_data,1,200)
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,

```

```

where      @w_id_local = 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 tran 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
if exists (select name from sysobjects where name = 'sp7tpcc_payment' )

```

```

go      drop procedure sp7tpcc_payment

create proc sp7tpcc_payment      @w_id      int,
                                @c_w_id     int,
                                @h_amount    int,
                                @d_id       tinyint,
                                @c_d_id     tinyint,
                                @c_id       int,
                                @c_last    char(16)

numeric(6,2),

= "

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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),

        @data      char(500),
        @c_data    char(500),
        @datetime  datetime,
        @w_ytd     numeric(12,2),
        @d_ytd     numeric(12,2),
        @cnt       smallint,
        @val       smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,

```

```

@c_id_local int
select @screen_data = "
begin tran 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 -
@c_h_amount,
           c_payment_cnt = c_payment_cnt + 1,
           c_ytd_payment = c_ytd_payment +
@c_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_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_data     = c_data,

where       @c_id_local = c_id
           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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update      customer
set         c_data = @c_data

where       c_id     = @c_id and
           c_w_id    = @c_w_id and
           c_d_id    = @c_d_id

select      @screen_data = substring
(@c_data,1,200)
end

```

```

-----
-- get district data and update year-to-date
-----
update      district
set         d_ytd = d_ytd +
@c_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 +
@c_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,

@c_id_local,

@c_w_id_local,

@datetime,

@c_h_amount,

@c_w_name + ' ' + @d_name)

commit tran 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

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   numeric(6,2),

                                @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_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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
    @d_id_local tinyint,
    @w_id_local int,
    @c_id_local int

select @screen_data = "

begin tran 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_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 -
    @h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    @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,
    @data = c_data,
    @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) +
substrning(@data, 1,
458)
-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substrning
(@c_data,1,200)
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 tran 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,
@d_name,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,

```

```

@e_state,
@c_phone,
@c_since,
@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data

go
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 numeric(6,2),
@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 numeric(12,2),

```

```

@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = ''

begin tran 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 -
@c_h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@c_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,
@data = c_data,
@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) +
substring(@data, 1,
458)

```

```

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update district
set d_ytd = d_ytd +
@c_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 +
@c_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 tran 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
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    numeric(6,2),

                              @d_id        tinyint,

```

```

= "

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 numeric(12,2),
        @c_balance   numeric(12,2),
        @c_discount  numeric(4,4),
        @data        char(500),
        @c_data      char(500),
        @datetime    datetime,
        @w_ytd       numeric(12,2),
        @d_ytd       numeric(12,2),
        @cnt         smallint,
        @val         smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = ""

begin tran p
-----
-- get payment date
-----
select    @datetime = getdate()

```

```

@c_id int,
@c_last char(16)

```

```

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,
          @data = c_data,

```



```

where      @c_id_local = c_id
           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) +
substring(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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_state = d_state, = d_city,
@d_zip   = d_zip,   = d_name,
@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 tran 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
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 numeric(6,2),
                             @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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = ''

begin tran 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 @w_id = @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_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@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,
@data = c_data,
@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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update district
set d_ytd = d_ytd +
@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 +
@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 tran 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
if exists (select name from sysobjects where name = 'sp12tpcc_payment' )

```

```

go drop procedure sp12tpcc_payment

create proc sp12tpcc_payment @w_id int,
                             @c_w_id int,
                             @h_amount numeric(6,2),
                             @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

```

```

select @screen_data = "
begin tran 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,
        @data           = c_data,

    where @c_id_local = c_id
          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) +

458)                substring(@data, 1,

-----
-- update customer info
-----
    update customer
    set c_data = @c_data

    where c_id = @c_id and
          c_w_id = @c_w_id and
          c_d_id = @c_d_id

    select @screen_data = substring
(@c_data,1,200)
    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 tran p

-----
-- return data to client
-----
    select @c_id,
           @c_last,

```

```

        @d_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
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 numeric(6,2),
                            @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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

```

```
select @screen_data = "
```

```
begin tran 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,
@data = c_data,
```

```
where @c_id_local = c_id
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) +
substr(@data, 1,
458)
```

```
-----
-- update customer info
-----
```

```
update customer
set c_data = @c_data
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id
```

```
select @screen_data = substring
(@c_data,1,200)
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 tran 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,

```

```

        @e_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data

go
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    numeric(6,2),

                                @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data      char(500),
        @c_data    char(500),
        @datetime  datetime,

```

```

        @d_year numeric(12,2),
        @cnt    smallint,
        @val    smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = ""

begin tran 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,
@data = c_data,

where
@c_id_local = c_id
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) +
458)
substring(@data, 1,

-----
-- update customer info
-----
update customer

```

```

set c_data = @c_data
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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 tran 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

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 numeric(6,2),
@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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = ''

begin tran p
-----
-- get payment date
-----
select @datetime = getdate()

```

```

if (@c_id = 0)
-----
-- 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 -
    @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,
    @data = c_data,
    @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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
       c_w_id = @c_w_id and
       c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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_id_local = d_id
                  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 tran 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,
        @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

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    numeric(6,2),
                                @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_phone    char(16),
        @c_since    datetime,
        @c_credit   char(2),
        @c_credit_lim numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,
        @val        smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = ""

begin tran 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

```

```

end          set          rowcount 0

-----
-- get customer info and update balances
-----
update      customer
set         @c_balance = c_balance -
@c_h_amount,
           c_payment_cnt      = c_payment_cnt + 1,
           c_ytd_payment      = c_ytd_payment +
@c_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,
           @data      = c_data,
           @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) +
substrning(@data, 1,
458)

-----
-- update customer info
-----
update      customer
set         c_data      = @c_data

where       c_id      = @c_id and
           c_w_id    = @c_w_id and
           c_d_id    = @c_d_id

select      @screen_data = substrning
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update      district
set         d_ytd      = d_ytd +
@c_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 +
@c_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 tran 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
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 numeric(6,2),
    @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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

```

```

select @screen_data = ""
begin tran 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_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @data = c_data,
    @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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
       c_w_id = @c_w_id and
       c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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 tran 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
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 numeric(6,2),
@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_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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int
select @screen_data = "
begin tran 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,
@data = c_data,

where @c_id_local = c_id
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) +
substring(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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_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 tran 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
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
numeric(6,2),
@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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),

```

```

@cnt smallint(12,2),
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@c_w_id_local int,
@c_id_local int

select @screen_data = "

begin tran 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_h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@c_h_amount,
@c_first = c_first,

```

```

@c_first = c_middle, = 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,
@data = c_data,

where @c_id_local = c_id
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) +
458) substring(@data, 1,

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

```

```

select      @screen_data = substring
(@c_data,1,200)
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,

```

```

commit tran p

declare @w_name char(20)

-----
-- 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
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   int,
                               @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 numeric(12,2),
       @c_balance   numeric(12,2),
       @c_discount  numeric(4,4),
       @data        char(500),
       @c_data      char(500),
       @datetime    datetime,
       @w_ytd       numeric(12,2),
       @d_ytd       numeric(12,2),
       @cnt         smallint,
       @val         smallint,
       @screen_data char(200),
       @d_id_local  tinyint,
       @w_id_local  int,
       @c_id_local  int

select @screen_data = "

begin tran 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 -
            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,
            @data        = c_data,
            @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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update      customer
set         c_data = @c_data
where       c_id = @c_id and
            c_w_id = @c_w_id and
            c_d_id = @c_d_id

select      @screen_data = substring
(@c_data,1,200)
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 +
            @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 tran 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
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    numeric(6,2),

                               @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,

```

```

        @e_credit_lim numeric(12,2),
        @c_balance    numeric(12,2),
        @c_discount   numeric(4,4),
        @data         char(500),
        @c_data       char(500),
        @datetime     datetime,
        @w_ytd        numeric(12,2),
        @d_ytd        numeric(12,2),
        @cnt          smallint,
        @val          smallint,
        @screen_data  char(200),
        @d_id_local  tinyint,
        @w_id_local  int,
        @c_id_local  int

select @screen_data = ""

begin tran 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,
        @data             = c_data,

        @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) +
                substring(@data, 1,
458)

```

```

-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
       c_w_id = @c_w_id and
       c_d_id = @c_d_id

select @screen_data = substr
(@c_data,1,200)
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 tran 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
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 numeric(6,2),
                           @d_id tinyint,
                           @c_d_id tinyint,

```

```

= "
@e_last int(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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

begin tran 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 -
            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,

```

```

            @data      = c_data,
            @c_id_local = 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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update      customer
set         c_data = @c_data

where       c_id     = @c_id and
            c_w_id    = @c_w_id and
            c_d_id    = @c_d_id

select      @screen_data = substring
            (@c_data,1,200)
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_name  = d_name,
            @d_id_local = d_id
            = @w_id and
            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
            = @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 tran 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
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 numeric(6,2),
@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),

```

```

@e_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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = "

begin tran 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
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,
@data = c_data,
@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) +
458)
substring(@data, 1,
-----
-- update customer info
-----
update customer
set c_data = @c_data
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id
select @screen_data = substring
(@c_data,1,200)
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 tran 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

```

```

if exists (select name from sysobjects where name = 'sp24tpcc_payment')
go
create proc sp24tpcc_payment @w_id int,
@c_w_id int,
@h_amount numeric(6,2),
@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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,

```

```

        @w_id_local int,
        @c_id_local int

select @screen_data = ''

begin tran 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,

```

```

        @e_street_2 = e_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,
        @data        = c_data,

        where
        @c_id_local = c_id
        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) +
                substring(@data, 1,
                458)

        -----
-- update customer info
-----
                update      customer
                set         c_data = @c_data

                where       c_id = @c_id and
                c_w_id = @c_w_id and
                c_d_id = @c_d_id

                select      @screen_data = substring
                (@c_data,1,200)
        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 tran 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
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  numeric(6,2),

                               @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_zstate   char(2),
        @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 numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,
        @val        smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

begin tran 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,
           @data       = c_data,

           @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) +
substr(@data, 1,
458)
-----
-- update customer info
-----
update customer
set c_data = @c_data
where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id
select @screen_data = substring
(@c_data,1,200)
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 tran 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_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data
go
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 numeric(6,2),
@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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),

```



```

@c_data      char(500),
@datetime   datetime,
@w_ytd      numeric(12,2),
@d_ytd      numeric(12,2),
@cnt        smallint,
@val        smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = ''

begin tran 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 -
@c_payment_cnt = c_payment_cnt + 1,
@c_ytd_payment = c_ytd_payment +

@c_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,
@data        = c_data,

@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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update      customer

```

```

set         c_data      = @c_data
where       c_id        = @c_id and
            c_w_id      = @c_w_id and
            c_d_id      = @c_d_id

select      @screen_data = substring

(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update      district
set         d_ytd      = d_ytd +
@c_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 +
@c_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 tran 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
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 numeric(6,2),
                                @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 numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,
        @val        smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

```

```

select @screen_data = "
begin tran 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,
        @data = c_data,
        @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) +
                                substring(@data, 1,
458)

-----
-- update customer info
-----
        update      customer
        set         c_data = @c_data

        where      c_id = @c_id and
                   c_w_id = @c_w_id and
                   c_d_id = @c_d_id

        select      @screen_data = substring
(@c_data,1,200)
        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_name = @d_name,
        @d_id_local = d_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 tran 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_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
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,
                                @h_amount    numeric(6,2),
                                @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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

select @screen_data = ''

begin tran 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_h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@c_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,
@data = c_data,
@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) +
substr(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update district
set d_ytd = d_ytd +
@c_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 +
@c_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 tran 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
if exists (select name from sysobjects where name = 'sp29tpcc_payment' )
    drop procedure sp29tpcc_payment
go

```

```

create proc sp29tpcc_payment
@c_d_id,
@c_w_id,
numeric(6,2),

= "

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 numeric(12,2),
        @c_balance numeric(12,2),
        @c_discount numeric(4,4),
        @data char(500),
        @c_data char(500),
        @datetime datetime,
        @w_ytd numeric(12,2),
        @d_ytd numeric(12,2),
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

select @screen_data = "

```

```

begin tran 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,
        @data = c_data,

        @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) +
        substring(@data, 1,
458)

-----
-- update customer info
-----
        update customer
set
    c_data = @c_data

        where
            c_id = @c_id and
            c_w_id = @c_w_id and
            c_d_id = @c_d_id

        select
            @screen_data = substring
(@c_data,1,200)

        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 tran p

-----
-- return data to client
-----
select
    @c_id,
    @c_last,
    @datetime,
    @w_street_1,

```

```

        @w_street_2,
        @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
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 int,
    numeric(6,2),
    @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 numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

```

```
select @screen_data = ''
```

```
begin tran 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 @wcount = (rowcount + 1) / 2

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,
@data = c_data,

@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) +
```

```
458)
```

```
substring(@data, 1,
```

```
-----
-- update customer info
-----
```

```

update customer
set c_data = @c_data

where c_id = @c_id and
c_w_id = @c_w_id and
c_d_id = @c_d_id

select @screen_data = substring
(@c_data,1,200)
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 tran 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_balance,
        @screen_data

go
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   numeric(6,2),
                                @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 numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),
        @cnt        smallint,

```

```

        @screen_data smallint,
        @d_id_local  tinyint,
        @w_id_local  int,
        @c_id_local  int

select @screen_data = "

begin tran 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,
@data = c_data,

where @c_id_local = c_id
      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) +
substrning(@data, 1,
458)

-----
-- update customer info
-----
update customer
set c_data = @c_data

where c_id = @c_id and
      c_w_id = @c_w_id and

```

```

      c_d_id = @c_d_id
select @screen_data = substring
(@c_data,1,200)
end

-----
-- get district data and update year-to-date
-----
update district
set d_ytd = d_ytd +
@c_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 +
@c_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,

@c_h_amount,

@w_name + ' ' + @d_name)

commit tran 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

-----
createstocklevproc.sql
-----
--
-- File: STOCKLEV.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.42
-- Copyright Microsoft, 2003
--
-- - Creates stock level transaction stored procedure
--
-- - Interface Level: 4.10.000
--

```

```

-----
use tpcc
go

if exists (select name from sysobjects where name = 'tpcc_stocklevel' )
    drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel@w_id      int,
                                     @d_id      tinyint,
                                     @threshold 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  < @threshold

option (order group)

go

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,
                                     @threshold smallint
as

declare    @o_id_low int,
           @o_id_high int

select    @o_id_low = (d_next_o_id - 20),

```

```

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  < @threshold

option (order group)

go

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,
                                     @threshold 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  < @threshold

option (order group)

go

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,
                                     @threshold 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  < @threshold

option (order group)

go

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,
                                     @threshold 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 < @threshold
option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

if exists (select name from sysobjects where name = 'sp7tpcc_stocklevel' )

```

```

go
    drop procedure sp7tpcc_stocklevel

create proc sp7tpcc_stocklevel @w_id int,
                              @d_id tinyint,
                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

if exists (select name from sysobjects where name = 'sp10tpcc_stocklevel' )
drop procedure sp10tpcc_stocklevel

go

create proc sp10tpcc_stocklevel @w_id int,

```

```

                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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 < @threshold

```

```

option (order group)

go

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,
                              @threshold 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 < @threshold

option (order group)

go

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,
                              @threshold 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   < @threshold

option (order group)

go

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,
                                     @threshold 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   < @threshold

option (order group)

go

if exists (select name from sysobjects where name = 'sp15tpcc_stocklevel' )

```

```

go      drop procedure sp15tpcc_stocklevel

create proc sp15tpcc_stocklevel      @w_id      int,
                                     @d_id      tinyint,
                                     @threshold 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   < @threshold

option (order group)

go

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,
                                     @threshold 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

```

```

         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   < @threshold

option (order group)

go

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,
                                     @threshold 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   < @threshold

option (order group)

go

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,
                                     @threshold 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   < @threshold

option (order group)

go

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,
                                   @threshold 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   < @threshold

option (order group)

```

```

go

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,
                                   @threshold 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   < @threshold

option (order group)

go

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,
                                   @threshold 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))

```

```

where   stock, order_line
        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   < @threshold

option (order group)

go

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,
                                   @threshold 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   < @threshold

option (order group)

go

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,

```

```

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 < @threshold

option (order group)

go

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,
                                   @threshold 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 < @threshold

option (order group)

go

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,
                                   @threshold 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 < @threshold

option (order group)

go

```

```

ol_o_id     = ol_o_id and
s_w_id     = ol_w_id and
s_i_id     = ol_i_id and
s_quantity < @threshold

option (order group)

go

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,
                                   @threshold 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 < @threshold

option (order group)

go

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,
                                   @threshold 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 < @threshold

option (order group)

go

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,
                                   @threshold 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 < @threshold

option (order group)

go

```

```

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,
                                   @threshold 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  < @threshold

option (order group)

go

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,
                                   @threshold 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  < @threshold

option (order group)

go

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,
                                   @threshold 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  < @threshold

option (order group)

go

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,
                                   @threshold 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  < @threshold

option (order group)

go

```





```

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\HarddiskVolume421
Locale           United States
Hardware Abstraction Layer  Version = "5.2.3790.1828
(srv03_sp1_rtm.050321-2020)"
User Name        Not Available
Time Zone        Pacific Daylight Time
Total Physical Memory 1,048,317.20 MB
Available Physical Memory 1,007.40 GB
Total Virtual Memory 1,018.22 GB
Available Virtual Memory 1,006.59 GB
Page File Space   2.00 GB
Page File        C:\pagefile.sys

```

[Hardware Resources]

[Conflicts/Sharing]

```

Resource  Device
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    Emulex LP9802 Fibre
Channel Host Adapter
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 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 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
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)

```

















IRQ 293	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 294 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 364	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 365 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 375	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 376 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 441	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 442 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 512	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 513 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 523	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 524 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 589	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 590 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 627	Emulex LP9802 Fibre Channel Host Adapter	OK
IRQ 660	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 661 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	

IRQ 671	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 672 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 737	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 738 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 808	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 809 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 819	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 820 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 885	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 886 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 956	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 957 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	
IRQ 967	Smart Array 6400 Controller (Non-Miniport)	OK
IRQ 968 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)	

IRQ 1033	Smart Array 6400 Controller (Non-Miniport)	OK	
IRQ 1034 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)		
IRQ 1104	Smart Array 6400 Controller (Non-Miniport)	OK	
IRQ 1105 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)		
IRQ 1115	Smart Array 6400 Controller (Non-Miniport)	OK	
IRQ 1116 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-OK)		
IRQ 1181	Smart Array 6400 Controller (Non-Miniport)	OK	
IRQ 1182 (Miniport)	Smart Array 6400 Controller U320 Expansion Module (Non-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	
0xC0040000-0xC0040FFF	PCI Serial Port	OK	
0xC0041000-0xC00413FF	Ethernet Controller	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	







```

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\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	
		3/30/2005 3:20 PM		bytes)	
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	
		3/30/2005 3:20 PM		bytes)	

[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&2A694D11&0&08
Last Reset	5/31/2005 7:46 AM
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	6/8/2005 8:13 AM
DHCP Lease Obtained	5/31/2005 8:13 AM
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&38E6BD97&0&08
Last Reset	5/31/2005 7:46 AM
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	6/8/2005 8:13 AM
DHCP Lease Obtained	5/31/2005 8:13 AM
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	Value
[00000003] RAS Async Adapter	
Adapter Type	Not Available
Product Type	RAS Async Adapter
Installed	Yes
PNP Device ID	Not Available
Last Reset	5/31/2005 7:46 AM
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	Value
[00000004] WAN Miniport (L2TP)	
Adapter Type	Not Available
Product Type	WAN Miniport (L2TP)
Installed	Yes
PNP Device ID	ROOT\MS_L2TPMINIPORT\0000
Last Reset	5/31/2005 7:46 AM
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	Value
[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	5/31/2005 7:46 AM
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 TypeWide Area Network (WAN)  
Product TypeWAN Miniport (PPPOE)  
Installed Yes  
PNP Device ID ROOT\MS\_PPPOEMINIPOINT\0000  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeDirect Parallel  
Installed Yes  
PNP Device ID ROOT\MS\_PTIMINIPOINT\0000  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeWAN Miniport (IP)  
Installed Yes  
PNP Device ID ROOT\MS\_NDISWANIP\0000  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection  
Installed Yes  
PNP Device ID Not Available  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection  
Installed Yes  
PNP Device ID Not Available  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection  
Installed Yes  
PNP Device ID Not Available  
Last Reset 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection  
Installed Yes  
PNP Device ID Not Available  
Last Reset 5/31/2005 7:46 AM  
Index 12  
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 [00000013] 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 5/31/2005 7:46 AM  
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 TypeNot Available  
Product TypeIntel(R) PRO/1000 MT Dual Port Network Connection  
Installed Yes  
PNP Device ID Not Available  
Last Reset 5/31/2005 7:46 AM  
Index 14  
Service Name E1000  
IP Address Not Available  
IP Subnet Not Available  
Default IP Gateway Not Available  
DHCP Enabled No





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 5/31/2005 7:46 AM  
 Index 24  
 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 [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 5/31/2005 7:46 AM  
 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 5/31/2005 7:46 AM  
 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 5/31/2005 7:46 AM  
 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 Type Not Available  
 Product Type Intel(R) PRO/1000 MT Dual Port Server Adapter  
 Installed Yes  
 PNP Device ID Not Available  
 Last Reset 5/31/2005 7:46 AM  
 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

[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 RSVIP 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 2.77 GB (2,978,754,560 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 7.67 GB (8,232,857,600 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 Q:

Description Local Fixed Disk

Compressed No

File System NTFS

Size 468.71 GB (503,270,338,560 bytes)

Free Space 468.63 GB (503,187,386,368 bytes)

Volume Name New Volume

Volume Serial Number B4DB5168

Drive X:

Description Local Fixed Disk

Compressed No

File System NTFS

Size 1.37 TB (1,509,811,023,872 bytes)

Free Space 1.36 TB (1,497,846,816,768 bytes)

Volume Name X\_Volume

Volume Serial Number 88A971EC

[Disks]

Item Value

[SCSI]

Item Value

[IDE]

Item Value

[Printing]

Name	Driver	Port Name	Server Name
b35-4053-a on prn-corp1	(from V-JASGOE02) in session 0		Xerox
Document Centre 255 PS	TS006		
b34-2525-b on red-prn-xrx	(from V-JASGOE02) in session 0		Xerox
Document Centre 255 PS	TS005		

[Problem Devices]

Device	PNP Device ID	Error Code
Not Available ACPI\HWP5003\2&DABA3FF&0		The drivers for this device are not installed.
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 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661471 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661512 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639005 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661359 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620821 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639159 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639130 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620833 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662841 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620702 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627436 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620635 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662570 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627572 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620643 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627576 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662825 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662700 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620585 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620709 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620770 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661341 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662815 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639045 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661336 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661533 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639154 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661520 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662567 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627429 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627469 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662945 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627480 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04627405 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04639122 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620931 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620644 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662816 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04538867 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620636 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04662631 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04620932 device are not installed. The drivers for this

Not Available ACPI\HWP1000\A56E04661421 device are not installed. The drivers for this

PCI Serial Port  
PCI\VEN\_103C&DEV\_1048&SUBSYS\_1223103C&REV\_02\5&4F5EBC7&0&00 The drivers for this device are not installed.

Ethernet Controller  
PCI\VEN\_1011&DEV\_0019&SUBSYS\_104F103C&REV\_41\5&4F5EBC7&0&08 The drivers for this device are not installed.

Not Available ACPI\IPI0001\0 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	PauseAccept
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	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
asynmac	RAS Asynchronous Media Driver	c:\win1828\system32\drivers\asynmac.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				
Driver	c:\win1828\system32\drivers\atmarpc.sys	Kernel			
	No Manual Stopped OK	Normal			
audstub	Audio Stub Driver				
Driver	c:\win1828\system32\drivers\audstub.sys	Kernel			
	Yes Manual Running OK	Normal			
b57nd	Broadcom NetXtreme Gigabit Ethernet				
Driver	c:\win1828\system32\drivers\b57xp64.sys	Kernel			
	Yes Manual Running OK	Normal			
beep	Beep	c:\win1828\system32\drivers\beep.sys	Kernel		
Driver	Yes System Running OK	Normal			
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				
changer	Changer	Not Available	Kernel Driver	No	System
	Stopped	OK	Ignore	No	No
clusdisk	Cluster Disk Driver				
Driver	c:\win1828\system32\drivers\clusdisk.sys	Kernel			
	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
cpqcissm	cpqcissm	c:\win1828\system32\drivers\cpqcissm.sys	Kernel Driver	Yes	Boot Running OK
	Normal	No	Yes		
cpqfcalm	cpqfcalm	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
crdisk	CRC Disk Filter Driver				
	c:\win1828\system32\drivers\crcdisk.sys	Kernel 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
	Yes				
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				
Driver	c:\win1828\system32\drivers\e1000645.sys	Kernel			
	Yes	Manual	Running	OK	Normal
	No	Yes			
elxstor	elxstor	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
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	Flypydisk	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				
hpciss	hpciss	Not Available	Kernel Driver	No	Disabled
	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				
Driver	c:\win1828\system32\drivers\hpqcissb.sys	Kernel			
	Yes	Boot	Running	OK	Normal
	No	Yes			
hpqcissd	Smart Array Controllers Non-Miniport Disk Driver				
Driver	c:\win1828\system32\drivers\hpqcissd.sys	Kernel			
	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\ipfldrv.sys	Kernel 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
	No	Yes			
ksecdd	KSecDD	c:\win1828\system32\drivers\ksecdd.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			

ksthunk	Kernel Streaming WOW64 Thunk Service				
Driver	c:\win1828\system32\drivers\ksthunk.sys	Kernel	Yes	Manual	Running OK
lp6nds35	lp6nds35	c:\win1828\system32\drivers\lp6nds35.sys	Kernel Driver	Yes	Boot Running OK
mnmdd	mnmdd	Not Available	Kernel Driver	No	System
modem	Modem	c:\win1828\system32\drivers\modem.sys	Kernel Driver	No	Manual Stopped OK
mouclass	Mouse Class Driver				
Driver	c:\win1828\system32\drivers\mouclass.sys	Kernel	Yes	System	Running OK
mountmgr	Mount Point Manager				
Driver	c:\win1828\system32\drivers\mountmgr.sys	Kernel	Yes	Boot	Running OK
mraid35x	mraid35x	Not Available	Kernel Driver	No	Disabled
mrxdav	WebDav Client Redirector				
System Driver	c:\win1828\system32\drivers\mrxdav.sys	File	Normal	No	Manual Stopped OK
mrxsmb	MRXSMB	c:\win1828\system32\drivers\mrxsmb.sys	File System Driver	Yes	System Running
msfs	Msfs	c:\win1828\system32\drivers\msfs.sys	File System Driver	Yes	System Running OK
mssmbios	Microsoft System Management BIOS Driver				
Driver	c:\win1828\system32\drivers\mssmbios.sys	Kernel	Yes	Manual	Running OK
mup	Mup	c:\win1828\system32\drivers\mup.sys	File System Driver	Yes	Boot Running OK
ndis	NDIS System Driver				
Driver	c:\win1828\system32\drivers\ndis.sys	Kernel Driver	Yes	Boot	Running OK
ndistapi	Remote Access NDIS TAPI Driver				
Driver	c:\win1828\system32\drivers\ndistapi.sys	Kernel	Yes	Manual	Running OK

ndisui	NDIS Usermode I/O Protocol				
Driver	c:\win1828\system32\drivers\ndisui.sys	Kernel	Yes	Manual	Running OK
ndiswan	Remote Access NDIS WAN Driver				
Driver	c:\win1828\system32\drivers\ndiswan.sys	Kernel	Yes	Manual	Running OK
ndproxy	NDIS Proxy	c:\win1828\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual Running OK
netbios	NetBIOS Interface				
System Driver	c:\win1828\system32\drivers\netbios.sys	File	Normal	No	Yes
netbt	NetBios over Tcpip				
Driver	c:\win1828\system32\drivers\netbt.sys	Kernel Driver	Yes	System	Running OK
nfrd960	nfrd960	Not Available	Kernel Driver	No	Disabled
npfs	Npfs	c:\win1828\system32\drivers\npfs.sys	File System Driver	Yes	System Running OK
ntfs	Ntfs	c:\win1828\system32\drivers\ntfs.sys	File System Driver	Yes	Disabled Running OK
null	Null	c:\win1828\system32\drivers\null.sys	Kernel Driver	Yes	System Running OK
partmgr	Partition Manager				
Driver	c:\win1828\system32\drivers\partmgr.sys	Kernel	Yes	Boot	Running OK
pci	PCI Bus Driver				
Driver	c:\win1828\system32\drivers\pci.sys	Kernel	Yes	Boot	Running OK
pciide	PCIIde	Not Available	Kernel Driver	No	Disabled
pcmcia	Pcmcia	c:\win1828\system32\drivers\pcmcia.sys	Kernel Driver	No	Disabled Stopped OK
pdcomp	PDCOMP	Not Available	Kernel Driver	No	Manual
pdframe	PDFRAME	Not Available	Kernel Driver	No	Manual

pdreli	PDRELI	Not Available	Kernel Driver	No	Manual
pdframe	PDRFRAME	Not Available	Kernel Driver	No	Manual
pptprnmiport	WAN Miniport (PPTP)				
Driver	c:\win1828\system32\drivers\raspptp.sys	Kernel	Yes	Manual	Running OK
processor	Processor Driver				
Driver	c:\win1828\system32\drivers\processr.sys	Kernel	Yes	Manual	Running OK
procinfo	ProcInfo	c:\win1828\system32\drivers\procinfo.sys	Kernel Driver	No	Manual Stopped OK
ptilink	Direct Parallel Link Driver				
Driver	c:\win1828\system32\drivers\ptilink.sys	Kernel Driver	Yes	Manual	Running OK
ql1080	ql1080	Not Available	Kernel Driver	No	Disabled
ql12160	ql12160	Not Available	Kernel Driver	No	Disabled
ql1240	ql1240	Not Available	Kernel Driver	No	Disabled
ql1280	ql1280	Not Available	Kernel Driver	No	Disabled
ql2200	ql2200	Not Available	Kernel Driver	No	Disabled
ql2300	ql2300	c:\win1828\system32\drivers\ql2300.sys	Kernel Driver	Yes	Boot Running OK
qlvika	qlvika	c:\win1828\system32\drivers\qlvika.sys	Kernel Driver	Yes	Auto Running OK
rasacd	Remote Access Auto Connection Driver				
Driver	c:\win1828\system32\drivers\rasacd.sys	Kernel Driver	Yes	System	Running OK
rasl2tp	WAN Miniport (L2TP)				
Driver	c:\win1828\system32\drivers\rasl2tp.sys	Kernel Driver	Yes	Manual	Running OK

rasppoe	Remote Access PPPOE Driver				
Driver	c:\win1828\system32\drivers\rasppoe.sys	Kernel	Yes	Manual	Running
		Normal	OK		
raspti	Direct Parallel				
Driver	c:\win1828\system32\drivers\raspti.sys	Kernel Driver	Yes	Manual	Running
		Normal	OK		
rdbss	Rdbss	c:\win1828\system32\drivers\rdbss.sys	File	Yes	System
System Driver			Running	OK	
	Normal	No	Yes		
rdpcdd	RDPCDD	c:\win1828\system32\drivers\rdpcdd.sys	Kernel	Yes	System
Driver			Running	OK	Ignore
	No	Yes			
rdpdr	Terminal Server Device Redirector Driver				
Driver	c:\win1828\system32\drivers\rdpdr.sys	Kernel Driver	Yes	Manual	Running
		Normal	OK		
rdpwd	RDPWD	c:\win1828\system32\drivers\rdpwd.sys	Kernel	Yes	Manual
Driver			Running	OK	Ignore
	No	Yes			
serial	Serial	c:\win1828\system32\drivers\serial.sys	Kernel	No	Auto
Driver			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	Stopped
		OK	Normal	No	Disabled
					No
slowmemory	Slow Memory Measurement				
Driver	c:\win1828\system32\drivers\slowmemory.sys	Kernel	No	Manual	Stopped
		Normal	OK		
srv	Srv	c:\win1828\system32\drivers\srv.sys	File	Yes	Manual
System Driver			Running	OK	
	Normal	No	Yes		
swenum	Software Bus Driver				
Driver	c:\win1828\system32\drivers\swenum.sys	Kernel	Yes	Manual	Running
		Normal	OK		
	No	Yes			
symc8xx	symc8xx	Not Available	Kernel Driver	No	Stopped
		OK	Normal	No	Disabled
					No
symmpi	symmpi	Not Available	Kernel Driver	No	Stopped
		OK	Normal	No	Disabled
					No
sym_hi	sym_hi	Not Available	Kernel Driver	No	Stopped
		OK	Normal	No	Disabled
					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						
Driver	c:\win1828\system32\drivers\tcpip.sys	Kernel Driver	Yes	System	Running	OK	Normal
			Normal				No
tdpipe	TDPIPE	c:\win1828\system32\drivers\tdpipe.sys	Kernel	No	Manual	Stopped	OK
Driver			Ignore	No	No		
tdtcp	TDTCP	c:\win1828\system32\drivers\tdtcp.sys	Kernel	Yes	Manual	Running	OK
Driver			Ignore	No	Yes		
termdd	Terminal Device Driver						
Driver	c:\win1828\system32\drivers\termdd.sys	Kernel	Yes	System	Running	OK	Normal
			Normal	No	Yes		
toside	TosIde	Not Available	Kernel Driver	No	Stopped	OK	Normal
		OK	Normal	No			Disabled
udfs	Udfs	c:\win1828\system32\drivers\udfs.sys	File	No	Disabled	Stopped	OK
System Driver			OK	No	No		
	Normal	No	No				
vgasave	VGA Display Controller.						
Driver	c:\win1828\system32\drivers\vgasave.sys	Kernel Driver	No	System	Stopped	OK	Ignore
			No				
viaide	ViaIde	Not Available	Kernel Driver	No	Stopped	OK	Normal
		OK	Normal	No			Disabled
							No
volsnap	Storage volumes						
Driver	c:\win1828\system32\drivers\volsnap.sys	Kernel	Yes	Boot	Running	OK	Normal
			Normal	No	Yes		
wanarp	Remote Access IP ARP Driver						
Driver	c:\win1828\system32\drivers\wanarp.sys	Kernel	Yes	Manual	Running	OK	Normal
			Normal	No	Yes		
wdica	WDICA	Not Available	Kernel Driver	No	Stopped	OK	Ignore
		OK	Ignore	No			Manual
							No
wlbs	Network Load Balancing						
Driver	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			
10/1/2002	Microsoft	netrasa.inf	Not Available		
ROOT\MS_PTMINIPORT\0000					
WAN Miniport (PPTP)	Yes	NET			
10/1/2002	Microsoft	netrasa.inf	Not Available		
ROOT\MS_PPTPMINIPORT\0000					
WAN Miniport (PPPOE)	Yes	NET			
10/1/2002	Microsoft	netrasa.inf	Not Available		
ROOT\MS_PPPOEMINIPORT\0000					
WAN Miniport (IP)	Yes	NET			
10/1/2002	Microsoft	netrasa.inf	Not Available		
ROOT\MS_NDISWANIP\0000					
WAN Miniport (L2TP)	Yes	NET			
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			
10/1/2002	(Standard system devices)	wave.inf	Not Available		
Available	ROOT\MEDIA\MS_MMMCI				
Legacy Audio Drivers	Yes	MEDIA			
10/1/2002	(Standard system devices)	wave.inf	Not Available		
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	Not Available
ROOT\LEGACY_WANARP\0000					
Available	Not Available	LEGACYDRIVER			
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_VOLSNAP\0000					

TDTCP Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_TDTCP\0000				
TCP/IP Protocol Driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_TCPIP\0000				
Slow Memory Measurement Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_SLOWMEMORY\0000				
sacdrv Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_SACDRV\0000				
RDPWD Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_RDPWD\0000				
RDPcDD Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_RDPcDD\0000				
Remote Access Auto Connection Driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_RASACD\0000				
qlvika Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_QLVIKA\0000				
ProcInfo Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_PROCINFO\0000				
Partition Manager Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_PARTMGR\0000				
Null Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NULL\0000				
NetBios over Tcpip Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NETBT\0000				
NDProxy Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NDPROXY\0000				
NDIS Usermode I/O Protocol Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NDISUIO\0000				
Remote Access NDIS TAPI Driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NDISTAPI\0000				
NDIS System Driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_NDIS\0000				
mountmgr Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_MOUNTMGR\0000				

ksecdd Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_KSECDD\0000				
IPSEC driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_IPSEC\0000				
IP Network Address Translator Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_IPNAT\0000				
Generic Packet Classifier Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_GPC\0000				
Fips Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_FIPS\0000				
FILEMON Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_FILEMON\0000				
dmload Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_DMLOAD\0000				
dmbot Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_DMBOT\0000				
CRC Disk Filter Driver Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_CRCDISK\0000				
Beep Available	Not Available	LEGACYDRIVER	Not Available	Not Available
ROOT\LEGACY_BEEP\0000				
AFD Available	Not Available	LEGACYDRIVER	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{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{235BD4BB-B766-4D21-B59A-9EA453F0AEB}				
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-EDFDECAB0E7}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C4399FF7-34F7-4FEC-ADE7-7A0B329FB9A7}				

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}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6E78BD1F-85A6-406C-AD5F-2B7568797113}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FEB39D35-6CA1-4189-BE32-909CCF70FBAC}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BEE3309C-CEFF-4E0B-905B-C58D69E7477F}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C535A213-B7C7-4948-8639-AB13CC61519A}				
Generic volume	Yes	VOLUME	5.2.3790.1828	
	10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{45CBB021-6E9D-4BB5-929C-D6623E14E3C0}				

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DE51FA1F-B63C-4009-828C-A79BEAC1136E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8C29FAF1-27C6-4D8E-96CC-83BCD4D8C201}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{616CF3EF-2779-48FD-8513-16868EC43F3A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DFD8936C-599B-470C-AB3E-AE3207FD199F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8210EF1E-0DEE-43D8-B169-18A799A4B0E3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E44DEFA8-4353-4B9E-B632-508CB5A04E18}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DE00C1F3-4D83-44E5-9F8A-21761ADA37C}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D63F9D93-976A-48D8-8E21-19E7377207EF}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F7443894-073B-461A-95F1-77371388EBFB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E2A0C596-0D0C-4AF3-867D-665990EE1729}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9BDCCEA-0175-4760-8179-53E8F60547C1}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FBF6EF36-3196-43A6-B704-FA31EF9BB986}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D53F9693-C148-453A-8D16-C2EB151A1A9B}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DD1280E6-BE62-4761-A5C4-C0C004C1CD70}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FD40779E-1C59-4A2F-A7D7-371F17B1091A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B768211C-307D-4F3E-B567-CF12215CBFBD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FD0DF07D-BF47-4C09-9917-5F62EC5854B9}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{781DFF10-671E-4EDF-A75A-317A043DA590}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D35E2CD4-7CE7-4A5B-9CCF-D7A6D1D72EAB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F3B51D41-9287-4BFF-BEE1-6930B88E2535}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{882AA3E0-3C5C-48AB-9A53-E7A1B33A0319}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA668A08-440D-4B8A-8EF9-43F02B0B4139}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A027ABD7-49A7-477A-B4E7-9D6EFB509154}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7D930E80-6A34-4C1E-AD12-D498FD57DD75}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B82464A7-2464-4437-B07D-5EAG65447512}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C162C4D6-C3AC-4448-8471-8A8793733145}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7F1DBD12-5ED6-4834-AFB1-64CA94AA8929}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DAF44947-B131-46AA-888C-6218E0A662DD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{54139985-D999-4BD7-9E91-16CF884B6109}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0BB2085F-5FD2-491C-B8D0-D9F5155AAF78}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{20B6ABEF-C3FA-4026-8AB7-367B74B1BDA5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DBFC67B8-E87A-4295-8F8A-3A9C2763A20C}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{44DBD0E3-A6DC-493C-B823-3DDAD631D27}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{602E48A2-05A7-4EB5-83B3-B752DADBE428}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{37FC8FFE-7058-474E-AF76-748EBCE93E34}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC40628A-55B0-4D23-8E66-3CF6C547FCDD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6D60FAA0-7FEC-4842-B584-8B307FDB2ADA}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1038C20F-440D-444B-8675-52F50D7F09F5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{682C736C-95F3-424C-8F65-41771783AB1F}			



Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8E236FF7  
-C43D-418B-9519-3996A90FD7E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{72EACEEA  
-2475-4760-AA8E-6AE7A45DCBC3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E9F172EC  
-7E39-4255-8F6F-51A7F1B6C877}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CEBC0C44  
-CDDA-46E3-9215-0E3155B6F60E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8588E6AB  
-B581-476D-8F28-43A1CA1E1863}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DFCB14FE  
-A3AF-4A2C-AB9E-F081417EB8B9}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8752F238  
-5348-4FE1-B193-F39A1B345AE8}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4D7614D9  
-A3A0-4AE0-A6FE-AF181DE425DF}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4932172A  
-6AA5-4DE8-B68F-47D2285F2B51}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5DB650D  
E-C3D6-4119-9633-8493A6C2B960}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C79C95EC  
-6192-4E45-8FE5-2C275813A3F3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BDA9C850  
-75C7-4445-83BB-4806541FEAF5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2DD04685  
-77D3-47AE-AC42-A1F17C3813B3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{70CE42E4  
-C642-4912-898A-DDC73966503A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E6D65CD  
7-9389-4944-AF69-6C347F3CDDDF}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{71797DD3  
-385D-4E14-8DC1-D6D8EE88C0FE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B4923F59  
-EBBB-483F-99CE-288FE901978E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A43F2313  
-194B-4DA7-A505-F3343D08DB2F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{54CE37F4  
-20AF-433F-85A2-33BA3CD4D722}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{69A50EAB  
-7FCC-4683-9810-93C1CD92854F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{34ACE7E9  
-D8DD-4794-AF2A-F72A0976F299}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BEDF9AA8  
-6C22-442C-B3D3-540A66DBCA6C}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B3736FFB  
-33A5-45B4-B2E1-272AB69B38AD}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7CFBEDA5  
-2E4B-4A1D-B248-31ABF360A39B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D19E822B  
-23B2-4355-8F51-07554E563814}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3CEFECD  
D-3A27-4A6C-9177-D72F811AAADDE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2B7EED67  
-57FB-45C7-87F1-3D0B29C1283A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{407A96C4  
-1020-43B3-BBC8-5F01C1FAA137}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A95B96A0  
-8466-4696-8624-420B63D14299}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B44C2D0F  
-F2FC-4F08-A48F-98F2891A07F1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4FD684C0  
-5444-4B2C-AC3D-EC74474B37F6}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AE399202  
-A77A-47C9-91DA-FA2E23418128}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC52D27  
4-E632-4A58-91D1-1F538D48F20B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AA60DBD  
B-484B-4B08-B03D-ED16BC00C71F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E1A71A5C  
-9C0E-4E56-8AFF-B0D6BB6D22C4}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B5B98188  
-42D0-4678-BF56-52620239CD20}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{806CDD6  
1-3327-41D6-BC0D-1DD020E6F106}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A6B6FCF7  
-DB47-40A5-88A2-2D7EEFC31C3A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CC48F84F  
-CB54-4235-B2A5-ED3EC8A4EA07}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\0FB287BA  
-A498-4062-9E94-581DA929E7E5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C6245D4E  
-9D79-484D-8ECA-0D51D779C5E8}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{E852FA88  
-FC2A-4181-BFA9-EBB7FD74C2D}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{54488647  
-AF77-49CB-BF91-20A4396E466B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{E686E295  
-214C-4585-B870-F3E73D8B2B8C}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{37B7C644  
-12A2-4364-9275-43255C7851A4}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{15E27E15  
-5228-46EB-9E84-AEF61F9199D5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{5CA038C3  
-17D3-4320-94B2-F015BB911B18}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{DF9CE67  
-3326-4ADB-8F84-0D020F8E5C2F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{40498B34  
-A214-4F04-A2A0-E8709BAC0BB5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{3E8D98D  
D-0AA7-4F20-81A5-B6C97A48D774}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{78B091C8  
-E7BC-425A-A6C2-D542F708F80E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{56A78655  
-E8CF-492E-A212-A3164F9CE022}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{A210CBEE  
-67D8-443C-B2E8-023987890FCC}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{1F02FC71  
-62B2-4FD8-ACCB-C0311B020943}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B82319C8  
-67EA-487F-AA83-BB8D4E8FE37B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{61D9D9E5  
-141E-454A-830F-18D5C8BFAE79}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9E504214  
-4A3D-4474-9429-2E49C8195248}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{335B966A  
-7CD2-4AE9-9536-EBE5F6FF3F02}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F0EBE4A7  
-C5BA-47C3-A5E3-2F362605A87D}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{23B78E75  
-D708-4745-8CA7-91FADA4D48E7}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{ABA17A79  
-90EA-41D8-A9AB-9791CD4992D6}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B8B1974E  
-8122-4943-A821-B8010F888A29}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9C7DF75E  
-7B65-4C08-91D3-6A40CE992508}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{8A0082E9  
-130C-4F7D-B978-710BC20E0F08}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{97CE6266  
-2CA9-4354-AD8A-1EB1666AC461}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2C0ACBF5  
-90E2-41EA-9FD1-D7E50DB2732D}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4CBE48C5  
-1DA7-44A4-BFF3-352DF26A324D}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{D4D22AE  
B-3F43-4923-A47A-1BC8D8E93AD9}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2ACA9AD  
6-CC20-4268-B790-414D7502ADF2}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{379A5DF1  
-5EFC-47E1-AF9D-67CA8C552108}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6EE8853B  
-1953-4ABC-926E-5879A3341969}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{5309E027  
-CD3D-4424-8BD0-89F39BDE0979}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4066F20A  
-9545-4830-B32E-A2912049428F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{67C6B34F  
-1F99-4B30-9DF5-B6718CB69FDB}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C26CAD3  
D-B7EF-4F2C-A68D-EE5B31665EEA}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B2FC77C5  
-C347-4E8E-B688-1B7686285135}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B30A82B5  
-785D-4B93-89FE-29912BE17A7F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{DE52B660  
-8066-4317-97BC-3009B72C27A9}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A0BD875  
D-662F-483E-AAFA-5684F7090BE1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C2EEB314  
-77AF-4EDF-8E65-F5789D2460A2}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{38880BC1  
-8C90-4233-8F66-9C1F2E404B1C}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{92CD3FE68  
-91EC-4034-893B-C6447E87A48A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FF831CB3  
-E5F7-494A-82C3-5BCE0DD6597F}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4ADEC7  
E-C3DD-43FF-A6B4-98C8901087AB}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0A538D9B  
-2527-45E6-A404-87503517D392}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{883E0D05  
-66E7-4617-A4B9-7D1F1DE6F76A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{294191A3  
-DBAA-4E01-99F2-A6C365F8556E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AEC4B696  
-A8B6-4956-AAEC-AC56271FFACF}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{77A03732  
-FBB4-49D1-A0CA-AC2312723DEE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A4ACD30F  
-58D9-4C51-979C-81AC624EBC43}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{918F75CB  
-640D-47A8-AFCF-29529955AC99}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{512B3D78  
-B8FB-403B-810E-84D83E8C1733}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DD5A285  
9-58C3-4ECE-8349-D69EFF0397D}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AD9BAA4  
8-38E6-48C3-8E05-73D0373231F5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F437A90C  
-B7DF-4D6B-A8FD-EFFD21E06611}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B9A4C542  
-F2FE-48AC-afb2-DB76952586E1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C3101670  
-7142-4F0F-B027-3140B00F8509}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{05217B06  
-FF20-46A3-81CA-B8189985A9F4}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F3CACB8  
D-3B19-4F07-86B4-B12A0FEFF952}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E6B276EC  
-A5BE-4D83-BF9B-11010B31A52A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F6C71882  
-5AD8-4777-9FA7-20BBB4CE3072}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{93226102  
-743B-49FC-8998-2AF0F11CABCA}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B4766BA4  
-FD70-4CAE-8CD1-3AA5CA7D6A48}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6339891D  
-94AE-4221-8439-1B776262BD48}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B633F27B  
-9CD0-42D7-A4FB-776866CF4F72}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E4890FA6  
-402E-48F0-9871-E58550483D7E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C51AF7FC  
-6626-48D2-BB35-C7861111CC30}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CBA65D95  
-6E10-4C2F-A0E3-E7BD4FA0052B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91545844  
-F80C-4018-A5DB-E7DD9F7E23FC}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5758FB79  
-C679-432E-BBC8-7E7453A1F2BB}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{505BBAC3  
-B61A-4826-9A3F-7635B99952BE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{798513D6  
-C213-48C9-BDB8-555729E7B390}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8076EAC5  
-E881-4569-A5AF-4E04AEF170B3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8160981D  
-CCB0-421A-915A-6FD07ACD2803}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{67A61F46  
-0296-47F5-ADCD-1A4F27FB67FB}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6FDBF413  
-0C71-4892-A070-82FFDFAF24BE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{384AFC74  
-07F8-4A0D-B15E-369594FB5F09}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{53F38754  
-E264-4A7E-891C-DB759AD9BE23}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{60E46C2C  
-3A0E-4A8D-ABB3-FBD567A43AB0}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{18677F4E  
-4584-4231-8026-276A2564B6B0}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9C513267  
-46CA-41A1-960C-7C96C2A584AC}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DD298045  
-46C5-49D5-B1B6-395D91C8A7D1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F26483D9  
-A473-43A3-B479-7917CE4EC812}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{304D86CC  
-F7CB-480C-AC24-E5235B5941D4}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{858A31E6  
-3551-4BBB-8C31-65967DF3FD4A}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{94BDC747  
-85FD-46A1-8D63-70F21CD4BBD3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1C7E55E7  
-D3A8-4C52-B4BA-C98BBA587CF3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5F6C9FD9  
-B62E-41F5-B8DA-415A44861051}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{249D9085  
-93CE-4E79-B8F3-ADE7EC568DF3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EF53E70D  
-5905-454E-9258-FED07E18EE81}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C7ABB563  
-4F69-4ED6-B1CA-3B7BCB976B47}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8EECEEE7  
-013E-4A96-82D6-AD7D56E46601}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A342B610  
-7DF6-4AE3-97C9-67845340B8CE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B27BE17D  
-A13A-4BE8-AAEB-F77F6DB022F1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{93511EC1  
-0C6A-4CAF-9757-506D8BEE84F7}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6C99C160  
-2CAD-45B4-941D-AF6BC8604EE5}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{812745F2  
-DB8B-46E5-A03C-C0DEBE47EC98}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{491D52FA  
-CC16-48B7-98D1-C6272D9CDBC0B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{97C92C93  
-1223-41D5-BA96-F1403BF03F94}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{75E89064  
-A9EE-44FA-87A7-5F90C77F3981}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2CD3BA9F  
-7219-42B7-90F4-31C3431D7227}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FBC33B90  
-C72C-4A26-A4E7-016114F9E476}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{06D43638  
-AFD6-4981-BC6D-CB8E01D00400}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B4D7C649  
-259C-4213-91FD-6AFBB2E86BD3}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{126E2B24  
-AAD5-491A-A2D3-85CCA7F75176}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{73305905  
-75C3-417F-94B3-19ABA3548390}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ACB0616C  
-1371-4B79-946A-E53978D4471C}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CF8079B7  
-DB37-422A-8C44-F052BF7876A1}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B769ED31  
-F902-423C-A020-949A8685C9AE}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E4EA7EDF  
-7798-476C-B0B6-ABAF0C3403B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{75629F39  
-4D13-459B-8555-91E6CAD47B5B}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D6CF47E  
-ECD5-4BFF-8608-800C76482271}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6BDFD55  
A-05F7-4A55-981C-BCE8AEB3B249}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AB03A871  
-EDFE-4190-845F-F8CDB4FC30E}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BF635B6A  
-7E41-40E5-985E-D37656131A86}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{44BA5575  
-D243-4614-ADFC-918B1DACB090}

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2C46FA4C-6123-4466-94E8-8C4842CB9BB1}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AB37D9B7-CE7-4FA7-8626-4CA80C8318FE}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5BB7A17C-C181-4EE0-97F1-3C153463E91E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{16043C25-302D-409F-9FD8-F93A68957300}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DE98764F-22BC-4063-AA93-364FF96B50DE}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{70B52B39-D3CC-4AAC-B5D4-A5CD1D7E5DD1}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C42FDB69-6C99-40BC-9368-F07D338F0961}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1345C491-4A42-421E-9F3A-AA3D83C122AE}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6CC88397-6E15-4DD8-B131-1D2405DCBAC2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B9255D9E-14BB-4913-8CE9-BD164A1525AE}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{45DAB1D-F-E142-4CFE-A9B2-49A3E6C64CAC}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{593C5422-1B4E-42FC-A988-A569A764CFA3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7C461FFA-39D1-4139-A30D-949ADCCAF73E}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BC52B889-00A1-4B2F-A9DF-A100BE68D083}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{860CDC76-6BBF-4FB6-88BA-143D5569888A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{42EF6231-845C-45B4-8778-502AC0896572}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{40E6CB22-481D-4AF0-AB70-0B2DE35503E2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{58E39605-315B-4D73-B672-603EE23896EB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B0149E34-F206-46CA-A6A8-F381F3F42407}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E8149B58-0ED1-4B1B-9295-27671D81DFC8}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E1AD78FE-3C6A-4409-8BBB-3DB42A58EDAD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0BF77F3E-F87E-49CC-BA9B-9D06DC5FE63E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3A9AAAF5-CCFD-4DFE-ADA2-9CD39EB7ECB5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E7880A2C-D826-46DD-9298-25C7C6AC67C8}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3E8E2B93-596D-4760-A2DF-F74DA3591004}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1F38EE29-0F3E-4410-939D-2BDDD3E7E10E}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B0B66619-4534-48B9-A0BB-0E212976527A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C61CF946-7E6E-4E50-86C9-D1F760EB56C0}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4970E4DE-7965-4044-8EBF-C1E0F3AD70EB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B9C3D369-00D2-4567-ABD0-3402935D3039}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E95F47DF-576C-4EB1-A2AD-A6188CA969E0}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{33809BFA-581F-4391-87A6-CED02E29BB5A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5ADAF1E0-8C22-4EE5-8AA2-DDBB7C8472C3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CB8FC619-5CA5-4286-AB41-A570165E2554}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{ABA361B3-59B9-439C-8916-DB81F25BA482}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CCF4335E-ED7D-4D02-B3C4-4AF051065272}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{327C49C7-509B-4483-9D12-7186435F632E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E46D49DE-C26E-4F32-9511-E49047E37D7F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A5F8AE21-57B1-4879-B2B1-5E0E996BC7EA}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4AD88810			
-F574-4DCF-A8A3-299FB28FA4F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{CFE74DA9			
-1860-4D8B-A340-E227BFEE3C9}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9AB7D104			
-BB3F-45C0-ABFB-E3C1BF31A5A6}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B590EE9A			
-4566-4D18-9AC1-C9D235C3605E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C334F66A			
-D076-4256-8C19-DBEC3A39EA5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{E634DD99			
-F84A-4000-BD83-05FE45B6F5C6}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{09B32430			
-B30A-4C11-9E40-FA649201D488}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6F076FBF			
-C0EE-421B-9B55-421ADBA1B1F3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{A11BBA33			
-B3D1-4FCC-93CD-53FB05C02018}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2E6AF897			
-8145-4071-AC70-9370377D9FBB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{157125E4			
-9429-418E-B6C1-683D9490CCA1}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{0EB18FCE			
-5BF3-4ACC-B7A4-E1110EBB431F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{9D96DB3			
7-8941-4638-97D1-A02576B3D4CA}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{AA827CB3			
-E4D1-4EDB-A9F0-3C6A917C01B0}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{D099DCB			
3-AEA4-4C24-B59C-6592D95147A2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{5B470FC7			
-19C7-4815-9F48-7DD5775DD108}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C929DBF6			
-188E-407B-9CD5-2DCD40D82737}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4972A1AB			
-805E-491D-BEE3-B77AF39F62A1}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6E150A67			
-58E0-48ED-BBB8-D2115BFB842E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{FC25AA08			
-CF12-4BE2-8DB2-E46B259800E2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{A9AD4D3			
C-1C23-45C2-A7AB-602F92E9BB7A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{D3CD501			
7-734D-4B0F-B798-03BB16338F74}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{AC60EE9E			
-5DFC-412B-A17A-649CD5CB3BEC}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{80826C5A			
-9360-4BA5-B43B-5F8247AB5925}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C75EE68A			
-3C63-4B67-BAE2-10FC25DBCE11}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4A5F9A56			
-25C2-4E5D-A8E6-AC2193EE6D61}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{1408F459			
-9522-4AFF-9BF6-24D8EC1F2AAD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{C70CEF60			
-0405-4D6C-9A9E-E11CBF3E4CA5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{3893AF6D			
-3D9C-4F55-AC05-968FEF45A2CF}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{ABA5EBE7			
-617C-4EFD-87EE-60EC1C21E33}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{B8B2360E			
-7C09-41FA-9041-7F7DF4014C70}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{D6F29342			
-2FD3-4E1D-BC71-216663CA74AD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{51773AC2			
-1348-4951-8FAE-2CFFB2C0AD31}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{41A22FC9			
-D8DE-46A0-9A12-FB311CDF0034}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F589B7DE			
-D183-4F7E-A928-A8189C6AF3A2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F32D90C7			
-D4BC-4191-B4FF-24871699FF71}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{E29A96D9			
-CEE3-4643-8F20-4085B5E6F591}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{41DCF375			
-6D60-447A-A86A-D2E1095B9D47}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{793CB841			
-E3C5-425D-A920-1555E196BDC0}			

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B5CF3DFE  
-0672-4E11-AD13-5948B033BBF6}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{15F928A1  
-01D8-4EFE-88DB-858EAE7D0FEB}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{10667AAB  
-490B-453D-8B2A-34A6684FF32B}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{172FBDC  
D-3939-4B74-86FD-1BB75D5049CF}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B9671B10  
-7FF2-4FCS-BE27-C7CCF04417B3}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{071315F9  
-3668-426A-9994-2126355D8E3F}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{85BFCFDB  
-64EE-41DC-8CB5-D84D70D29CC0}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E5F07441  
-D618-439E-9BDD-5AA597462C7D}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A340353B  
-D941-45BC-B310-FE99C39B4F39}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EDDB0A3  
7-7C6D-4234-8FAF-5C9AF3974B05}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EE20BE44  
-9A43-4728-9746-986E961BADDB8}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7F8962C0  
-77A0-4032-86ED-9C2B625B2585}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F42A70C6  
-AE5E-4B29-A1DB-217AC87735F2}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{630E07C1  
-2F3B-43A0-AEAB-9C91386E175F}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9BAEEFC9  
-8911-4952-96BC-0BA468665ACA}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D2B59559  
-1173-4964-BA42-AE91A64783E0}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BA899708  
-E912-4D2E-A9CC-3775B8520F48}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8AA26D77  
-6872-49AA-8AAC-6F7B1FDF346A}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3A20D4E3  
-27CB-464A-A19D-406AB001F4D0}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AD52146B  
-8396-4482-BAB1-501722322766}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{07C40507  
-D95A-4E43-AC04-D14116CC576E}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7A876698  
-A703-4C23-AC19-8EB146F180DA}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A371D2D  
C-8925-489C-A0CE-8ADDAFFD0D17}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E6DCEEE0  
-DF1E-427D-A70A-422A50F11F55}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{29C2C3F1  
-9F57-442B-B7A7-F99CCA8F635E}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{6D92BDA  
2-CC9E-4409-B513-39DA33BA7534}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4DBABDE  
C-AE47-49FD-A87A-722E87B81F1A}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D38E522F  
-F193-46E7-ABAE-2DE77BCB55E8}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D1079D86  
-4AFF-45A5-B773-16D0C7A77B5F}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{49D654FC  
-6139-4BB7-AE50-6D64938DA887}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{354D5470  
-064D-4984-90DC-D1B162E97E7A}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{932DC5F8  
-3BB0-4819-8ACC-334F071E8EAD}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{2CB76521  
-BD36-4B1A-8231-A53D328C5BB9}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{31243F96  
-E65F-4EDA-B755-ABA0C906A652}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{840D1529  
-9B2B-4AD7-9CA8-799B0BB0E9C9}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{497E1A56  
-21C1-40C2-B7B5-6F9051348DB2}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EEF51C0B  
-BDC0-4599-93BA-FCA66630DE40}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4F45F482  
-D015-4E34-8407-030C109805E4}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{92A305C6  
-FBEB-4E48-8E82-D0E875EBD573}

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{491D95D7-0AB7-4E86-A0FB-DE3D6661333}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B76320A5-9912-419D-A593-6CD8B41EF415}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A1941723-38D5-4AE4-B4A1-F9CDFDC25661}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DF5FFD82-7178-4859-95C3-61E19CC3676C}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F7D27D86-0AD1-4B4B-947C-4C2607731691}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{96C53B81-864C-4050-86AA-2E4796D928A2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{89740255-F9D2-4575-BE80-C7AC7CE551B5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{50817E93-FB8C-43B3-8DF0-78F03CD68893}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9C0B136B-2985-4E1C-A6E7-21D15C189D08}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1DE9017A-8B33-4083-AF4A-6F1FD2C196A3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{374EE174-46A5-46BF-A09B-697AD84B71CD}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{90FEC4F8-9ECB-49F4-8D7B-AF2B9CD490F0}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{C941C82E-4A4C-4B25-B315-8345A54F9B02}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{9D2EFAD E-19C4-48BF-8BAC-779CF4427612}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D413B762-735D-4102-9B9F-5589B6BD8BCC}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{35F2489A-EE77-4090-9D92-75BC7E5EEB7B}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E34D2782-A93B-427E-BB63-33BF2A7E62D8}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{52E0DAE5-A659-4865-A16E-9FD4284BC896}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC3EC235-A5A7-4346-8600-0974E00179FC}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5E19D7EF-6AE3-4D95-A007-7771806352EB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CD911F04-EA70-4E4E-AD53-A56DE6227E3F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{CE408A6B-54D6-425A-9497-DC7F2EC6E821}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1C496E88-93FA-4B2F-98F0-017BE58E9CEF}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DC3A7DB 1-F0B6-49A3-AAFC-3FB200ABC515}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4200608B-1D66-4F37-9CB7-413830B9E89E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{91329D8C-E905-4D56-B898-7C94D427CC8D}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{BFB390F1-ECF1-48B3-9C8F-9D8ACD21B9F2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{62FC3DA4-16B5-4E04-84D4-88F0D167AAD8}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1284416A-9B39-43E2-A7F9-0F555F84F54F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5881F5F0-171E-4CA6-9877-040268773B93}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0792A5AE-DC72-4ADF-BE05-8029616A92EF}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A86DCAB B-A6C8-4DFE-9823-D6D81CFC13AA}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{39E5D61A-7E4B-4061-A803-AEF570EC412E}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{33C0A94B-786A-4BCD-9EF9-9CFCEAF5CD7F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5C82AD1B-6672-46FA-823F-778CC06C5802}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3EAE77EC-4ED5-4A5E-9D3D-AEBF30714226}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1DC74AD 0-972F-4C1F-9BE7-3429796608F6}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D081C38E-82EE-4A1C-BE62-F3F3D04269A0}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4CA7E11A-2C75-44D2-AC5C-146AC4A028B3}			



Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FE6892DD  
-9FBE-4708-9C5E-7F98E9317429}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{3FEFB38C  
-5867-4F9A-8011-7318BA4EE66}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{02091B17  
-F86A-4B28-A3B9-3FAD7A369A6F}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EE563B06  
-FD74-47B9-B42F-97934678D92E}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{84893713  
-B85E-481C-AAAD-AF394C50887D}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5C812A97  
-091D-45AF-ADCF-EE2D4120B84D}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{DAD52D0  
E-6960-4B63-904C-3F6A15FB021A}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A3CE9A19  
-E65C-46BE-A9CD-1B1D18AFB52}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E5B9AE08  
-22D5-4382-919E-F6CCF6B5F408}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{FFBDB154  
-40B4-4AC2-915C-A9BACB2A1845}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5474817F  
-F348-44C5-B22F-936663E3B951}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8A5DC20  
D-40F1-4D87-937A-A5B9D21F2F26}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{188C447B  
-8FDD-4D81-BBE7-7E8F2B6D4382}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{A231F5FA  
-4FA1-45B7-AD51-CED9BA62C743}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{60B3DF6E  
-12C5-442E-847E-C7C311BBA29B}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{581A4DES  
-D59B-4D65-A353-EE66D6911A7C}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D9DFC1E  
E-E940-4248-B771-635F9B58A2D7}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{4042857C  
-9DB4-4E84-A4D9-4642EB084FD4}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{0D14F615  
-C227-451D-8D1A-4993D9967E6B}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{7C37CF4B  
-C89D-44CF-BD03-742C47AFFAB7}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8DC7F7E3  
-D045-4DAB-94D5-FDCB1266E409}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B782A703  
-F9E3-41F5-9F31-C80813A5B79}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{706E9F61  
-7BAB-43DD-BE45-28D2128292ED}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{8605E302  
-1DFB-4DC5-89DC-77891CD9775D}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{F41FBFDD  
-C655-467E-80C4-12562D677C21}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D4D50066  
-63E1-4108-8E01-664C28851C6}

Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D1417D33  
-5A69-46AD-B24F-4684EC6BAA2F}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{EFD62A3F  
-1418-42D9-9A82-08AE5AF7830E}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1FA21332  
-29E8-480E-A2C5-C74DC01C07AD}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{AD43BEC6  
-6D00-4837-9328-6A980FF11985}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{B045B042  
-61BF-4D5E-889A-C7FA2916D8B6}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1EF3D853  
-B9D8-4858-B2DB-8DA7587B5664}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{E09ECAC5  
-CC1B-41F8-A86A-D1FA0FC66DAB}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{1EC73114  
-212C-4BE2-851B-B10A516FB011}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{054EBEEF  
-02CA-4309-90B2-D02EC8BDD048}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{52C6F8E5  
-B727-47F9-9104-74BDAFA011E9}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{922CF545  
-AE93-4030-B1BA-FC5E5BCCFD66}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{5917F87D  
-B93F-4B0F-9F60-2D43F789B8B0}  
Generic volume Yes VOLUME 5.2.3790.1828  
10/1/2002 Microsoft volume.inf Not Available  
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION{D57BD1E  
1-3B68-41BC-A6BF-8324D058D795}

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{360A15D5			
-B857-464C-BD0B-7276EF857BD3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{20D27CD			
8-6029-411A-B52E-83CDE44A7A58}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{3CEA38A			
-7E2B-4BA4-B84E-BAE3A7BAEE75}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{0577BA90			
-F4E8-4BD9-AA15-7D5DA64B76A5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{48BB0A92			
-D7BF-4E97-BF39-21C59824CEF2}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{09D1F6A4			
-21CD-44AA-A8AE-F1458561521D}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{8E6F1AE6			
-DDD2-4C47-BD34-4F4FC704F53A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{DA59FF19			
-0855-4D72-96B7-9FD13F9FD605}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6ED13CC			
D-8E28-4900-98BE-1C9165553167}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{F8CAB606			
-F121-4B5C-AD3D-3C70BE92FA4F}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{394B651B			
-E00A-4983-BF42-36836B5D77BF}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6D646BC4			
-BE95-4F6B-9895-7B39A4FEA5ED}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{2969DAA3			
-52C3-40B9-868F-00C8D4D27948}			

Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{A200768A			
-50C8-441B-8395-AE30A6D315A9}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{DC525E36			
-4B0C-4F61-A5A8-70912ACA535D}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{8C57F57D			
-7ABE-49A5-89B2-CCA9DFC4F9F5}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{1576FC6C			
-E23C-445B-A309-B03F536D6DE7}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{004E368B			
-CECF-4BEB-9FE8-71CC441A5ACB}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{EA1838BC			
-41E6-430C-81A9-56FFF983A80A}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{8E086CAB			
-590B-4466-98EE-57BCF6A94969}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{89FD9A90			
-36AB-4F52-99E9-B72D8DF59250}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4ED7079E			
-92CB-447E-8692-BA011E01E5C3}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{6B33BA27			
-21CA-4E1E-946D-B79FCA5415D7}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{4264B3FE			
-0271-49E1-ABE6-22E83E20EA82}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPARTITION\{8CB4C345			
-9B6D-4FE6-8F35-A3A4771029D7}			
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#\{275A8610-7693-			
4E88-8F14-FC0AC0B18E48}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#\{80634918-058F-			
444D-99D4-935EF8899396}			
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#\{AA99A71-9232-			
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#\{BE54E31B-5E42-			
437A-A2E3-D48EB75C556D}			
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-D39FABC8F840}			
Generic volume	Yes	VOLUME	5.2.3790.1828
10/1/2002	Microsoft	volume.inf	Not Available
STORAGE\VOLUME\1&3735C57B&0&LDM#\{144532AA-5821-			
4B65-A141-79109FE1737F}			
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&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&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&2808  
 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  
 5F0&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  
 5F0&0&2808  
 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 scsdev.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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_\6&BD186B58&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&124050F  
 E&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&13  
 DF1870&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&32D965  
 2A&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&32D965  
 2A&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&13D  
 F1870&0&2808  
 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&19D  
 09E3&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&15575BF  
 4&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&15575BF  
 48&0&0000040000000000  
 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&1D  
 DED019&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&2E5959  
 D6&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&2E5959  
 D6&0&0000040000000000  
 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&1D  
 DED019&0&2808  
 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&787  
 45A4&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&2A4933  
 98&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&2A4933  
 98&0&0000040000000000  
 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&214  
 639D8&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&6DA58E  
 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&6DA58E  
 88&0&0000040000000000  
 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&214  
 639D8&0&2808  
 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&8336  
 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  
 D8&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  
 D8&0&0000040000000000  
 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&0000040000000000  
 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&2808

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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_ \6&33E4DACF&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&B37  
 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 scsdev.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\CD1  
 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\C3  
 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_00000000&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
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&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_00000000&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&822D
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&8279
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\A8  
 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  
 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\A0  
 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\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&1EB  
 0D7F0&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&1AB2D6  
 45&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&1AB2D6  
 45&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&1EB  
 0D7F0&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&1A9  
 D3CBE&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&20FA00  
 C7&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&20FA00  
 C7&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&132  
 20302&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&241FB56  
 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&241FB56  
 18&0&0000040000000000  
 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&132  
 20302&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&360  
 01D45&0&0808  
 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&RE  
 V\_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&A65  
 12F5&0&0808  
 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&RE  
 V\_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&7AB  
 2207&0&0808  
 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&2726FC3  
 48&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&2726FC3  
 48&0&0000040000000000  
 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&146  
 35F0&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&23A705  
 98&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&23A705  
 98&0&0000040000000000  
 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&146  
 35F0&0&2808  
 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&308  
 1E1E8&0&0808  
 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&3634E8  
 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&3634E8  
 C0&0&0000040000000000  
 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&20  
 DD8E1E&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&EE4CD6  
 8&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&EE4CD6  
 8&0&0000040000000000  
 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&20D  
 D8E1E&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&143  
 D29A6&0&0808  
 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  
 PCI bus Yes SYSTEM 5.2.3790.1828  
 10/1/2002 (Standard system devices) machine.inf Not  
 Available ACPI\HWP0002\84  
 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&E99B1A4&0&17C1  
 HP MSA1000 Yes SYSTEM 5.2.3790.1828  
 10/1/2002 Compaq scsidev.inf Not Available  
 SCSI\ARRAY&VEN\_COMPAQ&PROD\_MSA1000&REV\_4.32\6&E9  
 9B1A4&0&17C0



Emulex LP9802 Fibre Channel Host Adapter Yes  
 SCSIADAPTER 5.2.3790.1828  
 10/1/2002 Emulex pnpscsi.inf Not Available  
 PCI\VEN\_10DF&DEV\_F980&SUBSYS\_F98010DF&REV\_01\5&208  
 23031&0&08  
 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&397488  
 D1&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&397488  
 D1&0&0000040000000000  
 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&1B9  
 584B9&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&BA52D5  
 7&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&BA52D5  
 7&0&0000040000000000  
 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&1B9  
 584B9&0&2808  
 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&480  
 1F82&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&2EFC A0  
 26&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&2EFC A0  
 26&0&0000040000000000

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&261  
 56FBC&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&161D16  
 02&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&161D16  
 02&0&0000040000000000  
 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&261  
 56FBC&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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_6\6FFB230A&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  
 D32B&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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_6\61856D29B&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\_00000000&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\_00000000&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\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  
 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&293CF9F  
 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&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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_\6&1A490C64&0&07F0

QLogic QLA23xx PCI Fibre Channel Adapter Yes  
 8.2.2.10 4/16/2003 QLogic  
 SCSIADAPTER 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 scsdev.inf Not Available  
 SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
 V\_\6&1CFE3B12&0&07F0

QLogic QLA23xx PCI Fibre Channel Adapter Yes  
 8.2.2.10 4/16/2003 QLogic  
 SCSIADAPTER 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&848  
 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&0000040000000000  
 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&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&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&0000040000000000  
 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&8312  
 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  
 8&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  
 8&0&0000040000000000  
 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&8160  
 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&0000040000000000  
 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&8160  
 0BE56&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&349  
3B836&0&08  
PCI bus Yes SYSTEM 5.2.3790.1828  
10/1/2002 (Standard system devices) machine.inf Not  
Available ACPI\HWP0002\3A  
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&355FF3BE&0&17C1  
HP MSA1000 Yes SYSTEM 5.2.3790.1828  
10/1/2002 Compaq scsdev.inf Not Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_MSA1000&REV\_4.32\6&35  
5FF3BE&0&17C0  
Emulex LP9802 Fibre Channel Host Adapter Yes  
SCSIADAPTER 5.2.3790.1828  
10/1/2002 Emulex pnpscsi.inf Not Available  
PCI\VEN\_10DF&DEV\_F980&SUBSYS\_F98010DF&REV\_01\5&21  
D3DB6&0&08  
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 scsdev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
V\_16&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&8F8  
ADE6&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 scsdev.inf Not Available  
SCSI\PROCESSOR&VEN\_QLOGIC&PROD\_PSEUDO\_DEVICE&RE  
V\_16&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&355  
COD0F&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&28CBD6  
FB&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&28CBD6  
FB&0&0000040000000000  
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&16E  
A0F70&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&1C4DDF  
2D&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&1C4DDF  
2D&0&0000040000000000  
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&16E  
A0F70&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&31E  
E54F7&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&33F8103  
B&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&33F8103  
B&0&0000040000000000

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&353  
3F53F&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&1121A5  
ED&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&1121A5  
ED&0&0000040000000000  
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&353  
3F53F&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&28C  
A0570&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\_4.3  
2\6&2ABB8243&0&17C1  
HP MSA1000 Yes SYSTEM 5.2.3790.1828  
10/1/2002 Compaq scsdev.inf Not Available  
SCSI\ARRAY&VEN\_COMPAQ&PROD\_MSA1000&REV\_4.32\6&2A  
BB8243&0&17C0  
Emulex LP9802 Fibre Channel Host Adapter Yes  
SCSIADAPTER 5.2.3790.1828  
10/1/2002 Emulex pnpscsi.inf Not Available  
PCI\VEN\_10DF&DEV\_F980&SUBSYS\_F98010DF&REV\_01\5&381  
5DFDF&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 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&0000040000000000  
 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&0000040000000000  
 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&0000040000000000  
 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  
 Emulex LP9802 Fibre Channel Host Adapter Yes  
 SCSIADAPTER 5.2.3790.1828  
 10/1/2002 Emulex pnpscsi.inf Not Available  
 PCI\VEN\_10DF&DEV\_F980&SUBSYS\_F98010DF&REV\_01\5&FA5  
 C212&0&08  
 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 scsidesv.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 scsidesv.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&333D125&0&17C1  
 HP MSA1000 Yes SYSTEM 5.2.3790.1828  
 10/1/2002 Compaq scsidev.inf Not Available  
 SCSI\ARRAY&VEN\_COMPAQ&PROD\_MSA1000&REV\_4.32\6&33  
 3D125&0&17C0  
 Emulex LP9802 Fibre Channel Host Adapter Yes  
 SCSIADAPTER 5.2.3790.1828  
 10/1/2002 Emulex pnpscsi.inf Not Available  
 PCI\VEN\_10DF&DEV\_F980&SUBSYS\_F98010DF&REV\_01\5&32C  
 61491&0&08  
 PCI bus Yes SYSTEM 5.2.3790.1828  
 10/1/2002 (Standard system devices) machine.inf Not  
 Available ACPI\HWP0002\B  
 Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard  
 disk drives) disk.inf Not Available  
 SCSI\DISK&VEN\_HP&PROD\_LOGICAL\_VOLUME&REV\_2.32\7&A  
 D4AAFD&0&060  
 Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard  
 disk drives) disk.inf Not Available  
 SCSI\DISK&VEN\_HP&PROD\_LOGICAL\_VOLUME&REV\_2.32\7&A  
 D4AAFD&0&050  
 Disk drive Yes DISKDRIVE 5.2.3790.0 10/1/2002 (Standard  
 disk drives) disk.inf Not Available  
 SCSI\DISK&VEN\_HP&PROD\_LOGICAL\_VOLUME&REV\_2.32\7&A  
 D4AAFD&0&040  
 HP Virtual LUN Yes SYSTEM 5.2.3790.1828  
 10/1/2002 Compaq scsidev.inf Not Available  
 SCSI\OTHER&VEN\_COMPAQ&PROD SCSI\_COMMUNICATE&REV  
 \_CISS\7&AD4AAFD&0&000  
 Smart Array 6400 Controller Yes SCSIADAPTER  
 5.2.3790.1828 10/1/2002 Compaq  
 pnpscsi.inf Not Available  
 PCI\VEN\_0E11&DEV\_0046&SUBSYS\_409C0E11&REV\_01\6&266  
 093A2&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&29A  
 1C50A&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&137  
 A5C4D&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&137  
 A5C4D&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&38E  
 DBD97&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&2A6  
 94D11&0&08  
 PCI bus Yes SYSTEM 5.2.3790.1828  
 10/1/2002 (Standard system devices) machine.inf Not  
 Available ACPI\HWP0002\2  
 Not Available Not Available Not Available Not Available Not  
 Available Not Available Not Available ACPI\IPI0001\0  
 Ethernet Controller Not Available UNKNOWN Not Available Not  
 Available Not Available Not Available  
 PCI\VEN\_1011&DEV\_0019&SUBSYS\_104F103C&REV\_41\5&4F5  
 EBC7&0&08  
 PCI Serial Port Not Available UNKNOWN Not Available Not  
 Available Not Available Not Available Not Available  
 PCI\VEN\_103C&DEV\_1048&SUBSYS\_1223103C&REV\_02\5&4F5  
 EBC7&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  
Available Not Available Not Available ACPI\HWP1000\A56E04662909

Not Available Not Available Not Available Not Available Not Available Not Available Not Available  
Available Not Available Not Available ACPI\HWP1000\A56E04661474

Not Available Not Available Not Available Not Available Not Available Not Available Not Available  
Available Not Available 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\PNPOC0C\2&DABA3FF&0  
Microsoft ACPI-Compliant System Yes SYSTEM

5.2.3790.0 10/1/2002 Microsoft acpi.inf Not

Available ACPI\_HAL\PNPOC08\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 Not Available Not Available HTR\ROOT\0  
Not Available Yes Not Available 2:5.2 Not Available Not Available Not Available

Available Not Available Not Available b35-4053-a on prn-corp1 (from V-  
JASGOE02) in session 0

Not Available Yes Not Available 2:5.2 Not Available Not Available Not Available  
Available Not Available Not Available b34-2525-b on red-prn-xrx (from V-  
JASGOE02) in session 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>

lib C:\Program Files (x86)\SQLXML 4.0\bin\ <SYSTEM>

NUMBER\_OF\_PROCESSORS 64 <SYSTEM>

OS Windows\_NT <SYSTEM>

Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\Sy

stem32\Wbem;C:\Program Files\Microsoft SQL  
Server\80\Tools\Binn\C:\Program Files\Microsoft SQL

Server\90\Tools\Binn\C:\Program Files (x86)\Microsoft SQL  
Server\80\Tools\Binn\C:\Program Files (x86)\Microsoft SQL

Server\90\Tools\Binn\C:\Program Files (x86)\Microsoft SQL  
Server\90\Tools\Binn\VSShell\Common7\IDE\G:\Benchmarks\PerfTools\G:\

Benchmarks\SQLTools\G:\Benchmarks\SystemSetUpTools\  
<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>

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

TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL  
SERVICE

TEMP %USERPROFILE%\Local Settings\Temp NT  
AUTHORITY\NETWORK SERVICE

TEMP %USERPROFILE%\Local Settings\Temp NT  
AUTHORITY\NETWORK SERVICE

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

Z: \\hpwieshare\d\$ 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  
sms.exe Not Available 1148 11 409600 2826240

5/31/2005 8:08 AM Not Available Not Available Not

Available

csrss.exe c:\win1828\system32\csrss.exe 108 13

409600 2826240 5/31/2005 8:13 AM

5.2.3790.0 (srv03\_sp1\_rtm.030324-2048) 7.00 KB (7,168 bytes)

3/30/2005 3:20 PM

winlogon.exe c:\win1828\system32\winlogon.exe 308 13

409600 2826240 5/31/2005 8:13 AM

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 408 9

409600 2826240 5/31/2005 8:13 AM

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 420 9

409600 2826240 5/31/2005 8:13 AM

5.2.3790.0 (srv03\_sp1\_rtm.030324-2048) 15.00 KB (15,360

bytes) 3/30/2005 3:20 PM

svchost.exe c:\win1828\system32\svchost.exe 684 8

409600 2826240 5/31/2005 8:13 AM

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 788 8

409600 2826240 5/31/2005 8:13 AM

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 872 8

409600 2826240 5/31/2005 8:13 AM

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 896 8

409600 2826240 5/31/2005 8:13 AM

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 1000 8

409600 2826240 5/31/2005 8:13 AM

5.2.3790.1828 (srv03\_sp1\_rtm.050321-2020) 36.00 KB

(36,864 bytes) 3/30/2005 3:21 PM

spoolsv.exe c:\win1828\system32\spoolsv.exe 1860 8

409600 2826240 5/31/2005 8:14 AM

5.2.3790.1828 (srv03\_sp1\_rtm.050321-2020) 144.00

KB (147,456 bytes) 3/30/2005 3:21 PM

msdtc.exe c:\win1828\system32\msdtc.exe 1892 8

409600 2826240 5/31/2005 8:14 AM

2001.12.4720.1828 (srv03\_sp1\_rtm.050321-2020) 9.50 KB

(9,728 bytes) 3/30/2005 5:32 PM

svchost.exe c:\win1828\system32\svchost.exe 1988 8

409600 2826240 5/31/2005 8:14 AM

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 740 8

409600 2826240 5/31/2005 8:14 AM

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 360 8
409600 2826240 5/31/2005 8:14 AM
5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 36.00 KB
(36,864 bytes) 3/30/2005 3:21 PM
alg.exe c:\win1828\system32\alg.exe 2032 8
409600 2826240 5/31/2005 8:14 AM
5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 145.00
KB (148,480 bytes) 3/30/2005 3:20 PM
wmiprvse.exe c:\win1828\system32\wbem\wmiprvse.exe
2064 8 409600 2826240
5/31/2005 8:14 AM 5.2.3790.1828
(srv03_sp1_rtm.050321-2020) 718.50 KB (735,744 bytes)
3/30/2005 5:32 PM
csrss.exe c:\win1828\system32\csrss.exe 1836 13
409600 2826240 5/31/2005 8:36 AM
5.2.3790.0 (srv03_rtm.030324-2048) 7.00 KB (7,168 bytes)
3/30/2005 3:20 PM
winlogon.exe c:\win1828\system32\winlogon.exe 1692 13
409600 2826240 5/31/2005 8:36 AM
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 2988 8
409600 2826240 5/31/2005 8:36 AM
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 3092 8 409600
2826240 5/31/2005 8:37 AM 6.00.3790.1828
(srv03_sp1_rtm.050321-2020) 1.64 MB (1,720,320 bytes)
3/30/2005 3:20 PM
cmd.exe c:\win1828\system32\cmd.exe 1224 8
409600 2826240 5/31/2005 8:38 AM
5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 507.50
KB (519,680 bytes) 3/30/2005 3:20 PM
logon.scr c:\win1828\system32\logon.scr 764 4
409600 2826240 5/31/2005 8:46 AM
5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 703.00
KB (719,872 bytes) 3/30/2005 3:20 PM
taskmgr.exe c:\win1828\system32\taskmgr.exe 3524 13
409600 2826240 5/31/2005 9:24 AM
5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 427.50
KB (437,760 bytes) 3/30/2005 3:21 PM
sqlservr.exe c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe 4060 13 409600
2826240 5/31/2005 1:42 PM 2000.090.1114.60 built
by Administrator 67.10 MB (70,358,016 bytes)
5/24/2005 9:27 PM
msftesql.exe c:\program files\microsoft sql
server\mssql.1\mssql\bin\msftesql.exe 3612 8 409600
2826240 5/31/2005 1:46 PM 12.0.5104.1 274.70
KB (281,296 bytes) 3/4/2005 4:25 AM

```

```

wmiprvse.exe c:\win1828\system32\wbem\wmiprvse.exe
2820 8 409600 2826240
5/31/2005 3:14 PM 5.2.3790.1828
(srv03_sp1_rtm.050321-2020) 718.50 KB (735,744 bytes)
3/30/2005 5:32 PM

[Loaded Modules]

Name Version Size File Date Manufacturer Path
csrss 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\csrss.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
csrssv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 131.00
KB (134,144 bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\csrssv.dll
basesrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 156.50
KB (160,256 bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\basesrv.dll
winsrv 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 795.00
KB (814,080 bytes) 3/30/2005 3:21 PM Microsoft Corporation
c:\win1828\system32\winsrv.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
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
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
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
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
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
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
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 6.00.3790.1828 (srv03_sp1_rtm.050321-2020) 804.00
KB (823,296 bytes) 3/30/2005 3:21 PM 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				
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				
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				
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				
samilib	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\samilib.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	2001.12.4720.1828 (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	5.2.3790.0 (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				
printui	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	1.13 MB (1,186,304 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\printui.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				
adslrpc	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\adslrpc.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				
atl	3.00.2282	348.00 KB (356,352 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\atl.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				
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				

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				
cryptnet	5.2.3790.1828 (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				
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				
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				
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				
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				
msvcp60	6.10.2240.8 941.50 KB (964,096 bytes)		3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\msvcp60.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				
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				
scserv	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\scserv.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				
umpnpgmr	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\umpnpgmr.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				
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				
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				
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				

scedi	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\scedi.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				
wlbcctrl	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\wlbcctrl.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				
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				
dnssrslvr	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	125.50 KB (128,512 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\dnssrslvr.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
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
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
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
lmhsvc	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	40.00 KB (40,960 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\lmhsvc.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
rastls	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	426.00 KB (436,224 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\rastls.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
raschap	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	226.50 KB (231,936 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\raschap.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
aelupsvc	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	55.50 KB (56,832 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\aelupsvc.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
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
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
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
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
seclogon	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	47.00 KB (48,128 bytes)	3/30/2005 3:21 PM	Microsoft Corporation	c:\win1828\system32\seclogon.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
wuauerv	5.7.3790.1828 (srv03_sp1_rtm.050321-2020)	24.50 KB (25,088 bytes)	3/30/2005 5:34 PM	Microsoft Corporation	c:\win1828\system32\wuauerv.dll
wuaueng	5.7.3790.1828 (srv03_sp1_rtm.050321-2020)	3.08 MB (3,225,088 bytes)	3/30/2005 5:34 PM	Microsoft Corporation	c:\win1828\system32\wuaueng.dll

advpack	6.00.3790.1828 (srv03_sp1_rtm.050321-2020)	259.50 KB (265,728 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\advpack.dll
shfolder	6.00.3790.1828 (srv03_sp1_rtm.050321-2020)	42.50 KB (43,520 bytes)	3/30/2005 3:21 PM	Microsoft Corporation	c:\win1828\system32\shfolder.dll
winhttp	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	995.00 KB (1,018,880 bytes)	3/30/2005 7:45 AM	Microsoft Corporation	c:\win1828\winsxs\ia64_microsoft.windows.winhttp_6595b64144cc1df_5.1.3790.1828_x-ww_a36ef564\winhttp.dll
mshpacha	5.2.3790.0 (srv03_rtm.030324-2048)	80.00 KB (81,920 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\mshpacha.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
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
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
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
ntlsapi	5.2.3790.0 (srv03_rtm.030324-2048)	14.50 KB (14,848 bytes)	3/30/2005 3:20 PM	Microsoft Corporation	c:\win1828\system32\ntlsapi.dll

spoolsv	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	144.00 KB (147,456 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\spoolsv.exe				
spoolss	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	246.50 KB (252,416 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\spoolss.dll				
localspl	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	943.50 KB (966,144 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\localspl.dll				
cnbjmon	5.2.3790.1224 (dnsvr\skatari).040514-1058)	100.00 KB (102,400 bytes)	3/21/2005 3:09 PM	Microsoft Corporation
c:\win1828\system32\cnbjmon.dll				
pljmon	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	42.50 KB (43,520 bytes)	3/21/2005 3:18 PM	Microsoft Corporation
c:\win1828\system32\pljmon.dll				
tcpmon	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	143.00 KB (146,432 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\tcpmon.dll				
wsnmp32	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	128.50 KB (131,584 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\wsnmp32.dll				
tcpmib	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	47.00 KB (48,128 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\tcpmib.dll				
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				
mgmtapi	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	47.00 KB (48,128 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\mgmtapi.dll				
snmpapi	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	61.00 KB (62,464 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\snmpapi.dll				
usbmon	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	49.00 KB (50,176 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\usbmon.dll				
winmr	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	47.50 KB (48,640 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\winmr.dll				
win32spl	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	276.00 KB (282,624 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\win32spl.dll				
inetpp	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	214.50 KB (219,648 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\inetpp.dll				
icmp	5.2.3790.0 (srv03_rtm.030324-2048)	2.50 KB (2,560 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\icmp.dll				
ps5ui	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	394.00 KB (403,456 bytes)	4/1/2005 6:20 PM	Microsoft Corporation
c:\win1828\system32\spool\drivers\ia64\3\ps5ui.dll				

msdtc	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	9.50 KB (9,728 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\msdtc.exe				
msdtctm	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	2.94 MB (3,087,360 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\msdtctm.dll				
msdtcprx	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	1.27 MB (1,329,152 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\msdtcprx.dll				
mtxclu	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	203.00 KB (207,872 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\mtxclu.dll				
msdtclog	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	223.00 KB (228,352 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\msdtclog.dll				
xolehlp	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	18.00 KB (18,432 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\xolehlp.dll				
resutils	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	154.00 KB (157,696 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\resutils.dll				
mtxoci	2001.12.4720.1828 (srv03_sp1_rtm.050321-2020)	314.50 KB (322,048 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\mtxoci.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				
regsvc	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	240.00 KB (245,760 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\regsvc.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				
alg	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	145.00 KB (148,480 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\alg.exe				
wmiprvse	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	718.50 KB (735,744 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\wbem\wmiprvse.exe				
faultrep	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	250.00 KB (256,000 bytes)	3/30/2005 3:20 PM	Microsoft Corporation
c:\win1828\system32\faultrep.dll				

wmiprov	5.2.3790.1828 (srv03_sp1_rtm.050321-2020)	503.00 KB (515,072 bytes)	3/30/2005 5:32 PM	Microsoft Corporation
c:\win1828\system32\wbem\wmiprov.dll				
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				
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	6.00.3790.1828 (srv03_sp1_rtm.050321-2020)	3.49 MB (3,658,240 bytes)	3/30/2005 3:21 PM	Microsoft Corporation
c:\win1828\system32\shdocvw.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				
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				
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				
browsecl	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\browsecl.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

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

cmd       5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 507.50
KB (519,680 bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\cmd.exe

logon     5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 703.00
KB (719,872 bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\logon.scr

taskmgr  5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 427.50
KB (437,760 bytes) 3/30/2005 3:21 PM Microsoft Corporation
c:\win1828\system32\taskmgr.exe

utildll  5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 70.00 KB
(71,680 bytes) 3/30/2005 3:21 PM Microsoft Corporation
c:\win1828\system32\utildll.dll

sqlservr 2000.090.1114.60 built by Administrator 67.10 MB (70,358,016
bytes) 5/24/2005 9:27 PM Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe

msvcpc80 8.00.41204.17 1.38 MB (1,447,424 bytes) 4/8/2005
2:19 PM Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\msvcpc80.dll

msvcr80 8.00.41204.17 1.43 MB (1,495,552 bytes) 4/8/2005
2:19 PM Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\msvcr80.dll

opends60 2000.090.1116.00 29.50 KB (30,208 bytes) 4/7/2005
11:32 AM Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\opends60.dll

msvcr80 8.00.50215.37 1.40 MB (1,472,000 bytes)
3/31/2005 8:50 PM Microsoft Corporation
c:\win1828\winsxs\ia64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_
8.0.50215.4645_x-ww_e7089338\msvcr80.dll

instapi 2000.090.1116.00 80.50 KB (82,432 bytes) 4/7/2005
11:32 AM Microsoft Corporation c:\program files\microsoft sql
server\90\shared\instapi.dll

sqlevn70 2000.090.1116.00 1.48 MB (1,550,336 bytes) 4/7/2005
12:28 AM Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\resources\1033\sqlevn70.rll

sqlos 2000.090.1116.00 11.00 KB (11,264 bytes) 4/7/2005
11:32 AM Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlos.dll

mscoree 2.0.50215.37 (beta2.050215-3700) 693.00 KB (709,632
bytes) 3/31/2005 11:32 PM Microsoft Corporation
c:\win1828\system32\mscoree.dll

security 5.2.3790.0 (srv03_rtm.030324-2048) 7.00 KB (7,168 bytes)
3/30/2005 3:21 PM Microsoft Corporation
c:\win1828\system32\security.dll

```

```

msfte     12.0.5104.1 6.70 MB (7,029,456 bytes) 3/4/2005 4:25 AM
Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\msfte.dll

dbghelp  5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 2.15 MB
(2,257,920 bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\dbghelp.dll

qlvpl    1.2 439.50 KB (450,048 bytes) 5/21/2005 3:52 PM
QLogic Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\qlvpl.dll

sqlncli  2000.090.1116.00 5.44 MB (5,699,288 bytes) 4/7/2005
1:41 PM Microsoft Corporation c:\win1828\system32\sqlncli.dll

sqlncldr 2000.090.1116.00 130.71 KB (133,848 bytes) 4/7/2005
1:40 PM Microsoft Corporation c:\win1828\system32\sqlncldr.rll

msftepxy 12.0.5104.1 79.70 KB (81,616 bytes) 3/4/2005 4:25 AM
Microsoft Corporation
c:\progra~1\micro~1\mssql.1\mssql\bin\msftepxy.dll

msftesql 12.0.5104.1 274.70 KB (281,296 bytes) 3/4/2005 4:25 AM
Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\msftesql.exe

msftepxy 12.0.5104.1 79.70 KB (81,616 bytes) 3/4/2005 4:25 AM
Microsoft Corporation c:\program files\microsoft sql
server\mssql.1\mssql\bin\msftepxy.dll

cimwin32 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 4.81 MB
(5,046,272 bytes) 3/30/2005 5:32 PM Microsoft Corporation
c:\win1828\system32\wbem\cimwin32.dll

framedyn 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 536.50
KB (549,376 bytes) 3/30/2005 5:32 PM Microsoft Corporation
c:\win1828\system32\wbem\framedyn.dll

ntevt    5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 691.00
KB (707,584 bytes) 3/30/2005 5:32 PM Microsoft Corporation
c:\win1828\system32\wbem\ntevt.dll

provthrd 5.2.3790.1828 (srv03_sp1_rtm.050321-2020) 631.00
KB (646,144 bytes) 3/30/2005 5:32 PM Microsoft Corporation
c:\win1828\system32\wbem\provthrd.dll

msvcirt 7.0.3790.0 (srv03_rtm.030324-2048) 161.50 KB (165,376
bytes) 3/30/2005 3:20 PM Microsoft Corporation
c:\win1828\system32\msvcirt.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

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

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 5.31.23.1224 1.38 MB (1,451,008 bytes) 3/30/2005 3:21 PM
Microsoft Corporation c:\win1828\system32\riched20.dll

```

[Services]

Display Name	Name	State	Start Mode	Service
Type	Path	Error Control	Start Name	Tag ID
Application Experience Lookup Service	AeLookupSvc	Running	Auto	Share Process
netsh	Normal	LocalSystem	0	0
alerter	Alerter	Stopped	Disabled	Share Process
Application Layer Gateway Service	ALG	Running	Manual	Normal
Application Management Process	AppMgmt	Stopped	Manual	Share
ASP.NET State Service	aspnet_state	Stopped	Manual	Own
e.exe	Normal	NT AUTHORITY\NetworkService	0	0
Windows Audio Process	AudioSrv	Stopped	Disabled	Share
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share Process
netsh	Normal	LocalSystem	0	0
Computer Browser Process	Browser	Running	Auto	Share
Indexing Service	CISvc	Stopped	Disabled	Share
ClipBook	ClipSrv	Stopped	Disabled	Own Process
.NET Runtime Optimization Service v2.0.50215_X86	clr_optimization_v2.0.50215_32	Stopped	Manual	Own Process
.NET Runtime Optimization Service v2.0.50215_I64	clr_optimization_v2.0.50215_64	Stopped	Manual	Own Process
xe	Ignore	LocalSystem	0	0
COM+ System Application Process	COMSysApp	Stopped	Manual	Own

Cryptographic Services Process	CryptSvc	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
DCOM Server Process Launcher	DcomLaunch	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k LocalSystem 0				
dcomlaunch	Normal	LocalSystem 0		
Distributed File System Process	Dfs	Stopped	Manual	Own Normal
c:\win1828\system32\dfsrv.exe LocalSystem 0				
DHCP Client	Dhcp	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k networkservice NT AUTHORITY\NetworkService 0				
Logical Disk Manager	Administrative Service		dmadmin	Stopped Normal
c:\win1828\system32\dmadmin.exe /com LocalSystem 0				
Logical Disk Manager Process	dmserver	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
DNS Client	Dnscache	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k networkservice NT AUTHORITY\NetworkService 0				
Error Reporting Service Process	ERSvc	Running	Auto	Share Ignore
c:\win1828\system32\svchost.exe -k winerr LocalSystem 0				
Event Log	Eventlog	Running	Auto	Share Normal
c:\win1828\system32\services.exe LocalSystem 0				
COM+ Event System Process	EventSystem	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Help and Support	helpsvc	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Human Interface Device Access	HidServ	Stopped	Disabled	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
HTTP SSL	HTTPFilter	Stopped	Manual	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				
IAS Jet Database Access Process	IASJet	Stopped	Manual	Share Normal
c:\win1828\syswow64\svchost.exe -k iasjet LocalSystem 0				
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled	Own Normal
c:\win1828\system32\imapi.exe LocalSystem 0				
Intersite Messaging Process	IsmServ	Stopped	Disabled	Own Normal
c:\win1828\system32\ismserv.exe LocalSystem 0				
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				

Server	lanmanserver	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Workstation Process	lanmanworkstation	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
License Logging	LicenseService	Stopped	Disabled	Own Normal
c:\win1828\system32\llssrv.exe NT AUTHORITY\NetworkService 0				
TCP/IP NetBIOS Helper Process	LmHosts	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k localservice NT AUTHORITY\LocalService 0				
Messenger	Messenger	Stopped	Disabled	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Disabled	Own Normal
c:\win1828\system32\mnmsrvc.exe LocalSystem 0				
Distributed Transaction Coordinator	MSDTC	Running	Auto	Share Normal
c:\win1828\system32\msdtc.exe NT AUTHORITY\NetworkService 0				
msftesql	msftesql	Running	Manual	Own Normal
"c:\program files\microsoft sql server\mssql.1\mssql\bin\msftesql.exe" -s:mssql.1 -f:mssqlserver LocalSystem 0				
Windows Installer Process	MSIExec	Stopped	Manual	Share Normal
c:\win1828\system32\msiexec.exe /v LocalSystem 0				
Microsoft Search	MSSEARCH	Stopped	Auto	Share Normal
"c:\program files\common files\system\mssearch\bin\mssearch.exe" LocalSystem 0				
SQL Server (MSSQLSERVER)	MSSQLSERVER	Stopped	Manual	Share Normal
c:\program files\microsoft sql server\mssql.1\mssql\bin\sqlservr.exe" -smssqlserver LocalSystem 0				
MSSQLServerADHelper	MSSQLServerADHelper	Stopped	Manual	Own Normal
c:\program files\microsoft sql server\90\shared\sqladhlp90.exe" LocalSystem 0				
Network DDENetDDE	NetDDE	Stopped	Disabled	Share Normal
c:\win1828\system32\netdde.exe LocalSystem 0				
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share Normal
c:\win1828\system32\netdde.exe LocalSystem 0				
Net Logon	Netlogon	Stopped	Manual	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				
Network Connections	Netman	Running	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				

Network Location Awareness (NLA) Process	Nla	Running	Manual	Share Normal
c:\win1828\system32\svchost.exe -k LocalSystem 0				
File Replication Process	NtFrs	Stopped	Manual	Own Ignore
c:\win1828\system32\ntfrs.exe LocalSystem 0				
NT LM Security Support Provider	NtLmSsp	Running	Manual	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				
Removable Storage Process	NtmsSvc	Stopped	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Plug and Play\PlugPlay	Running	Auto	Share	Process Normal
c:\win1828\system32\services.exe LocalSystem 0				
IPSEC Services Process	PolicyAgent	Running	Auto	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				
Protected Storage	ProtectedStorage	Running	Auto	Share Normal
c:\win1828\system32\lsass.exe LocalSystem 0				
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Remote Access Connection Manager	RasMan	Stopped	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Remote Desktop Help Session Manager	RDSSMgr	Stopped	Manual	Share Normal
c:\win1828\system32\sessmgr.exe LocalSystem 0				
Routing and Remote Access	RemoteAccess	Stopped	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs LocalSystem 0				
Remote Registry	RemoteRegistry	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k regsvc NT AUTHORITY\LocalService 0				
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual	Share Normal
c:\win1828\system32\locator.exe NT AUTHORITY\NetworkService 0				
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Share Normal
c:\win1828\system32\svchost.exe -k rpcss NT AUTHORITY\NetworkService 0				
Resultant Set of Policy Provider	RSOPProv	Stopped	Manual	Share Normal
c:\win1828\system32\rsopprov.exe LocalSystem 0				
Special Administration Console Helper	sacsvr	Running	Manual	Share Normal
c:\win1828\system32\svchost.exe -k netsvcs 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
Process	c:\win1828\system32\scardsvr.exe		Ignore	NT
LocalService	0			
AUTHORITY\Task Scheduler	Schedule	Running	Auto	Share
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Secondary Logon	seclogon	Running	Auto	Share
Process	c:\win1828\system32\svchost.exe -k netsvcs		Ignore	
LocalSystem	0			
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
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	
LocalSystem	0			
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Ignore	LocalSystem 0
Print Spooler	Spooler	Running	Auto	Own Process
Process	c:\win1828\system32\spoolsv.exe		Normal	
LocalSystem	0			
SQL Browser	SQLBrowser	Stopped	Manual	Own Process
Process	"c:\program files\microsoft sql server\90\shared\sqlbrowser.exe"		Normal	LocalSystem 0
SQL Server Agent (MSSQLSERVER)	SQLSERVERAGENT	Stopped	Manual	Own Process
Process	"c:\program files\microsoft sql server\mssql.1\mssql\bin\sqlagent90.exe" -i mssqlserver		Normal	LocalSystem 0
SQL Writer	SQLWriter	Stopped	Manual	Own Process
Process	"c:\program files\microsoft sql server\90\shared\sqlwriter.exe"		Normal	LocalSystem 0
Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled	Share Process
Process	c:\win1828\system32\svchost.exe -k imgsvc		Normal	NT AUTHORITY\LocalService 0
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual	Own Process
Process	c:\win1828\system32\svchost.exe -k swprv		Normal	LocalSystem 0
Performance Logs and Alerts	SysmonLog	Stopped	Auto	Own Process
Process	c:\win1828\system32\smlogsvc.exe		Normal	NT Authority\NetworkService 0
Telephony	TapiSrv	Stopped	Manual	Share Process
Process	c:\win1828\system32\svchost.exe -k tapisrv		Normal	LocalSystem 0
Terminal Services	TermService	Running	Manual	Share Process
Process	c:\win1828\system32\svchost.exe -k termsvc		Normal	LocalSystem 0

Themes	Themes	Stopped	Disabled	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
Distributed Link Tracking Client	TrkWks	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
Terminal Services Session Directory	Tssdis	Stopped	Disabled	Own Process
Process	c:\win1828\system32\tssdis.exe		Normal	LocalSystem 0
Uninterruptible Power Supply	UPS	Stopped	Manual	Own Process
Process	c:\win1828\system32\ups.exe		Normal	NT AUTHORITY\LocalService 0
Virtual Disk Service	vds	Stopped	Manual	Own Process
Process	c:\win1828\system32\vds.exe		Normal	LocalSystem 0
Volume Shadow Copy	VSS	Stopped	Manual	Own Process
Process	c:\win1828\system32\vssvc.exe		Normal	LocalSystem 0
Windows Time	W32Time	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k localSERVICE		Normal	NT AUTHORITY\LocalService 0
WebClient	WebClient	Stopped	Disabled	Share Process
Process	c:\win1828\system32\svchost.exe -k localSERVICE		Normal	NT AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share Process
Process	c:\win1828\system32\svchost.exe -k localSERVICE		Normal	NT AUTHORITY\LocalService 0
Windows Management Instrumentation	wimgmt	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Ignore	LocalSystem 0
Windows Management Instrumentation Driver Extensions	Wmi	Stopped	Manual	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
WMI Performance Adapter WmiApSrv	Wbem	Stopped	Manual	Own Process
Process	c:\win1828\system32\wbem\wmiaprsrv.exe		Normal	LocalSystem 0
Automatic Updates	wuauerv	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
Wireless Configuration	WZCSVC	Running	Auto	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0
Network Provisioning Service	xmlprov	Stopped	Manual	Share Process
Process	c:\win1828\system32\svchost.exe -k netsvcs		Normal	LocalSystem 0

[Program Groups]

Group Name	Name	User Name	Default User	Default User
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\Analysis Services	All Users:Microsoft SQL Server 2005 CTP\Analysis Services	All Users		
Microsoft SQL Server 2005 CTP\Configuration Tools	All Users:Microsoft SQL Server 2005 CTP\Configuration Tools	All Users		
Microsoft SQL Server 2005 CTP\Performance Tools	All Users:Microsoft SQL Server 2005 CTP\Performance 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	Startup
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
DWQueuedReporting	"v:\progra~1\common~1\micro~1\dw\dwtrig20.exe" -t	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"
Bitmap Image	mspaint.exe

[Windows Error Reporting]

Time	Type	Details
------	------	---------

[Internet Settings]

[Internet Explorer]

[ Following are sub-categories of this main category ]

[Summary]

Item	Value
------	-------

[File Versions]

File	Version	Size	Date	Path	Company
------	---------	------	------	------	---------

[Connectivity]

Item	Value
------	-------

[Cache]

[ Following are sub-categories of this main category ]

[Summary]

Item	Value
------	-------

[List of Objects]

Program File	Status	CodeBase
--------------	--------	----------

[Content]

[ Following are sub-categories of this main category ]

[Summary]

Item	Value
------	-------

[Personal Certificates]

Issued To	Issued By	Validity	Signature	Algorithm
-----------	-----------	----------	-----------	-----------

[Other People Certificates]

Issued To	Issued By	Validity	Signature	Algorithm
-----------	-----------	----------	-----------	-----------

[Publishers]

[ Following are sub-categories of this main category ]

[Security]

Zone	Security Level
------	----------------

**Microsoft SQL Server 2005 Startup Parameters**

```
sqlservr.exe" -c -x -T205 -T661 -T677 -T697 -T827 -T834 -T1211 -T3502 -
T8011 -T8012 -T8018 -T8020 -T8193 -T8710 -T8744
where
rem ** -c - Run as console app
rem ** -x - Disable stats
rem ** -T205 - Print reason for recompile
rem ** -T661 - Disable the ghost record removal process
rem ** -T677 - Disable bunch of checks in AM. Available in
chk/fire, not golden
rem ** -T697 - Make schema latch superlatch
rem ** -T827 - Display latch promotions
rem ** -T834 - Large pages
rem ** -T1211 - Disable lock escalation
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"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Bootstrap]
"BootstrapDir"="C:\Program Files\Microsoft SQL Server\90\Setup
Bootstrap\"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Bootstrap\MSIRefCount]
"Uninstall"="{11111111-1111-1111-1111-111111111111},{EA83A5A6-0295-
4EEB-8408-BCC919C13841},{DAC5E237-5F9A-4A65-9426-418E6955E38B}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node0]
"CpuMask"=hex:0f,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node1]
"CpuMask"=hex:f0,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node2]
"CpuMask"=hex:00,0f,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node3]
"CpuMask"=hex:00,f0,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node4]
"CpuMask"=hex:00,00,0f,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node5]
"CpuMask"=hex:00,00,f0,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node6]
"CpuMask"=hex:00,00,00,0f,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node7]
"CpuMask"=hex:00,00,00,f0,00,00,00,00

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node8]
"CpuMask"=hex:00,00,00,00,0f,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node9]
"CpuMask"=hex:00,00,00,00,0f,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node10]
"CpuMask"=hex:00,00,00,00,00,0f,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node11]
"CpuMask"=hex:00,00,00,00,00,0f,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node12]
"CpuMask"=hex:00,00,00,00,00,00,0f,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node13]
"CpuMask"=hex:00,00,00,00,00,00,0f,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node14]
"CpuMask"=hex:00,00,00,00,00,00,00,0f

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node15]
"CpuMask"=hex:00,00,00,00,00,00,00,70

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node16]
"CpuMask"=hex:00,00,00,00,00,00,00,80

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\SQL
Browser]
"AnalysisServiceListener"=dword:00000000
"SsrplListener"=dword:00000000
"Contexts"=dword:00000004
"Threads"=dword:00000004
"msmdlocalpath"="C:\Program Files\Microsoft SQL
Server\90\Shared\msmdlocal.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\SQL
Browser\Servers]
"SQLServer"="1434"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools]

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\ClientSetup]
"SQLPath"="C:\Program Files\Microsoft SQL Server\90\Tools"

"checksum"=hex:38,38,33,66,66,66,30,65,62,38,32,62,30,34,37,62,31,30,35,
31,36,\

38,32,31,30,30,63,64,36,66,33,37,66,38,63,32,38,65,65,30,38,65,61,66,37,3
7,\

35,64,36,33,31,32,62,35,32,66,61,63,33,39,62,32,64,37,33,61,65,65,63,63,6
5,\

66,61,64,65,62,64,63,37,34,38,39,66,62,39,32,37,64,63,31,32,64,66,30,34,3
0,\

35,35,36,61,65,66,39,62,32,31,61,36,36,39,62,61,66,61,66,61,36,36,38,39,3
3,\

35,66,32,61,38,64,66,33,31,33,38,65,63,34,31,39,63,66,64,34,61,31,65,64,6
6,\

61,64,65,35,33,61,31,61,66,64,31,62,61,34,34,61,38,36,36,30,65,39,39,62,6
3,\

30,37,64,61,38,33,62,00
"SQLBinRoot"="C:\Program Files\Microsoft SQL Server\90\Shared\"
"Path"="C:\Program Files\Microsoft SQL Server\90\Tools\bin\"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\ClientSetup\CurrentVersion]
"Language"=dword:00000409
"CurrentVersion"="9.00.1116"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\Service Manager]
"Services"=hex(7):3b,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\Tools\Setup]
"Version"="9.00.1116"
"Language"=dword:00000409
"ProductCode"="{DAC5E237-5F9A-4A65-9426-418E6955E38B}"
"FeatureList"="Client_Components=3 Server_tools=3 Connectivity=3
SQL_Tools90=3 SQL_ComputerManagement=3 SQL_WorkBench=3
SQL_Profiler90=3"
"PatchLevel"="9.00.1116"
"checksum"=hex:38,38,33,66,66,66,30,65,62,38,32,62,30,34,37,62,31,30,35,
31,36,\

38,32,31,30,30,63,64,36,66,33,37,66,38,63,33,61,61,39,63,34,32,62,31,30,3
9,\

```

35,66,66,65,33,35,34,64,65,65,33,35,61,37,35,61,31,37,62,36,32,38,66,36,65,\

65,34,36,34,62,31,62,35,31,36,38,32,30,34,35,36,33,37,34,32,36,35,30,33,61,\

63,31,66,66,62,37,61,63,35,39,30,38,31,63,64,36,32,30,31,32,34,32,30,36,34,\

62,35,63,31,38,65,66,66,65,38,35,31,66,34,37,34,39,66,37,32,35,61,38,32,31,\

32,61,37,65,62,61,35,66,34,36,63,63,64,63,34,30,66,65,32,32,34,61,65,30,65,\

37,39,39,61,35,37,36,00

"Edition"="Beta Edition (64-bit)"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Shortcut]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\Tools\Shortcut\SQL Configuration Manager]

"1033"=""

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\ExceptionMessageBox]

"ShowFeedbackButton"=dword:00000001

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names\SQL]

"MSSQLSERVER"="MSSQL.1"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1]

@="MSSQLSERVER"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\CPE]

"ErrorDumpDir"="C:\\Program Files\\Microsoft SQL Server\\MSSQL.1\\MSSQL\\LOG\\"

"EnableErrorReporting"=dword:00000000

"CustomerFeedback"=dword:00000000

"TimeOfReporting"=dword:0000003c

"CollectorParameters"="SQL MSSQLSERVER"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{01C6B350-12C7-11CE-BD31-00AA004BBB1F}]

@="infosoft.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{12CE94A0-DEFB-11D2-B31D-00600893A857}]

@="chsbrkr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{1305C72E-2073-452E-A72D-667032B3B4FE}]

@="msir5jp.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{1680E7C3-9430-4A51-9B82-1E7E7AE5258}]

@="chtbrkr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{16BC5CE4-2C78-4CB9-80D5-386A68CC2B2D}]

@="danlr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{173C97E2-AEBE-437C-9445-01B237ABF2F6}]

@="langwrbk.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{188D6CC5-CB03-4C01-912E-47D21295D77E}]

@="langwrbk.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{20036404-F1AF-11D2-A57F-006052076F32}]

@="ruslr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{20036414-F1AF-11D2-A57F-006052076F32}]

@="ruslr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{23A9C1C3-3C7A-4D2C-B894-4F2864F9DAD6}]

@="ptblr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{25B7FD48-5404-4BEB-9D80-B6982AF404FD}]

@="ptblr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{2A6EB050-7F1C-11CE-BE57-00AA0051FE20}]

@="infosoft.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{41B9BE05-B3AF-460C-BF0B-2CDD44A093B1}]

@="xmlfilt.dll"

"ThreadingModel"="Both"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{52F1DEDC-C51B-11D2-9A7C-00004F6888ED}]

@="msfte.dll"

"ThreadingModel"="Both"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{5D5F3A69-620C-4952-B067-4D0126BB6086}]

@="ptslr.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{66B37110-8BF2-11CE-BE59-00AA0051FE20}]

@="infosoft.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{68C6D24E-77E0-11D5-8528-00C04F68155C}]

@="msfte.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{6D36CE10-7F1C-11CE-BE57-00AA0051FE20}]

@="infosoft.dll"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{6DE705A5-6467-43DC-9CE3-CCFE08B3F645}]

@="nls400.dll"

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{737FC9B5-BC66-4953-B7DB-B79A72A592A5}]
@="korwbrkr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{73D009FB-7D05-45AC-94C2-EDD1029942D4}]
@="langwrbk.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{83BC7EF7-D27B-4950-A743-0F8E5CA928F8}]
@="danlr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{860D28D0-8BF4-11CE-BE59-00AA0051FE20}]
@="infosoft.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{8DF412D1-62C7-4667-BBEC-38756576C21B}]
@="trklr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{9478F640-7F1C-11CE-BE57-00AA0051FE20}]
@="infosoft.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{95679344-48B8-42CD-9B79-2F5784D70258}]
@="langwrbk.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{97E8EFC5-42DA-43ED-9CEC-84FDE3620F1A}]
@="korwbrkr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{9EA69E16-AD5E-43ED-B90E-73D58771D3F6}]
@="nls400.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{A9FFAE02-D834-47B3-8434-0D75DC0804BD}]
@="msir5jp.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{B0516FF0-7F1C-11CE-BE57-00AA0051FE20}]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{B8713269-2D9D-4BF5-BF40-2615D75723D8}]
@="lrpolish.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{C731071F-AC80-11D1-8DF3-00C04FB6EF4F}]
```

```
@="msfte.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}]
@="msfte.dll"
"ThreadingModel"="Both"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{CA665B09-4642-4C84-A9B7-9B8F3CD7C3F6}]
@="lrpolish.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{CA22CF4-59FE-11D1-BBFF-00C04FB97FDA}]
@="thawbrkr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{CEDC01C7-59FE-11D1-BBFF-00C04FB97FDA}]
@="thawbrkr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{D4171BC4-90BE-4F70-8610-DAB1C17F063C}]
@="ptslr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{D5FCDD7E-DBFF-473F-BCCD-3AFD1890EA85}]
@="ptblr.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{D99F7670-7F1A-11CE-BE57-00AA0051FE20}]
@="infosoft.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{E0CA5340-4534-11CF-B952-00AA0051FE20}]
"ThreadingModel"="Both"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{EEED4C20-7F1B-11CE-BE57-00AA0051FE20}]
@="infosoft.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{F07F3920-7B8C-11CF-9BEB-00AA004B9986}]
```

```
@="offilt.dll"
"ThreadingModel"="Both"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\CLSID\{FD86B5D0-12C6-11CE-BD31-00AA004BBB1F}]
@="infosoft.dll"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters]
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\ascx]
@="{E0CA5340-4534-11CF-B952-00AA0051FE20}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\asm]
@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\asp]
@="{E0CA5340-4534-11CF-B952-00AA0051FE20}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\aspx]
@="{E0CA5340-4534-11CF-B952-00AA0051FE20}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\bat]
@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\c]
@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\cmd]
@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"
```



@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\.xlc]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\.xls]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\.xlt]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\.xml]

@="{41B9BE05-B3AF-460C-BF0B-2CDD44A093B1}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020810-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020811-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020820-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020821-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020900-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{00020906-0000-0000-C000-000000000046}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{25336920-03F9-11CF-8FD0-00AA00685190}]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{59850400-6664-101B-B21C-00AA004BA90B}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{5E941D80-BF96-11CD-B579-08002B30BFEB}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{64818D10-4F9B-11CF-86EA-00AA00B929E8}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{64818D11-4F9B-11CF-86EA-00AA00B929E8}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{7F73B8F6-C19C-11D0-AA66-00C04FC2EDDC}]

@="{E0CA5340-4534-11CF-B952-00AA0051FE20}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{87DB1ADA-AA39-11D1-829F-00A0C906B239}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{89BCB7A4-6119-101A-BCB7-00DD010655AF}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{8C9E8E1C-90F0-11D1-BA0F-00A0C906B239}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{95876EB0-90F0-11D1-BA0F-00A0C906B239}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{961C1130-89AD-11CF-88A1-00AA00685190}]

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{9ED4692C-90F0-11D1-BA0F-00A0C906B239}]

@="{C7310720-AC80-11D1-8DF3-00C04FB6EF4F}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{BD70C020-2D24-11D0-9110-00004C752752}]

@="{E0CA5340-4534-11CF-B952-00AA0051FE20}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\Filters\CLSID\{EA7BAE70-FB3B-11CD-A903-00AA00510EA3}]

@="{F07F3920-7B8C-11CF-9BE8-00AA004B9986}"

[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSearch\GlobalAdminBlob]

"Blob0"=hex:00,00,00,80,a0,02,00,00,06,00,15,00,46,00,54,00,45,00,5f,00,52,00,\

65,00,74,00,61,00,69,00,6c,00,54,00,72,00,61,00,63,00,69,00,6e,00,67,00,50,\

00,61,00,74,00,68,00,00,00,08,00,38,00,43,00,3a,00,5c,00,50,00,72,00,6f,00,\

67,00,72,00,61,00,6d,00,20,00,46,00,69,00,6c,00,65,00,73,00,5c,00,4d,00,69,\

00,63,00,72,00,6f,00,73,00,6f,00,66,00,74,00,20,00,53,00,51,00,4c,00,20,00,\

53,00,65,00,72,00,76,00,65,00,72,00,5c,00,4d,00,53,00,53,00,51,00,4c,00,2e,\

00,31,00,5c,00,4d,00,53,00,53,00,51,00,4c,00,5c,00,4c,00,4f,00,47,00,5c,00,\

00,00,14,00,46,00,54,00,45,00,5f,00,53,00,51,00,4c,00,44,00,75,00,6d,00,70,\

00,65,00,72,00,44,00,75,00,6d,00,70,00,44,00,69,00,72,00,00,00,08,00,38,00,\

43,00,3a,00,5c,00,50,00,72,00,6f,00,67,00,72,00,61,00,6d,00,20,00,46,00,69,\

```

00,6c,00,65,00,73,00,5c,00,4d,00,69,00,63,00,72,00,6f,00,73,00,6f,00,66,00,
\
74,00,20,00,53,00,51,00,4c,00,20,00,53,00,65,00,72,00,76,00,65,00,72,00,5c
.\
00,4d,00,53,00,53,00,51,00,4c,00,2e,00,31,00,5c,00,4d,00,53,00,53,00,51,00
.\
4c,00,5c,00,4c,00,4f,00,47,00,5c,00,00,00,12,00,46,00,54,00,45,00,5f,00,53,
\
00,51,00,4c,00,44,00,75,00,6d,00,70,00,65,00,72,00,46,00,6c,00,61,00,67,00
.\
73,00,00,00,08,00,06,00,30,00,78,00,30,00,38,00,30,00,30,00,00,15,00,4
6,\
00,54,00,45,00,5f,00,53,00,51,00,4c,00,44,00,75,00,6d,00,70,00,65,00,72,00
.\
46,00,75,00,6c,00,6c,00,50,00,61,00,74,00,68,00,00,00,08,00,3e,00,43,00,3a
.\
00,5c,00,50,00,72,00,6f,00,67,00,72,00,61,00,6d,00,20,00,46,00,69,00,6c,00
.\
65,00,73,00,5c,00,4d,00,69,00,63,00,72,00,6f,00,73,00,6f,00,66,00,74,00,20,
\
00,53,00,51,00,4c,00,20,00,53,00,65,00,72,00,76,00,65,00,72,00,5c,00,39,00
.\
30,00,5c,00,53,00,68,00,61,00,72,00,65,00,64,00,5c,00,5c,00,73,00,71,00,6c
.\
00,64,00,75,00,6d,00,70,00,65,00,72,00,2e,00,65,00,78,00,65,00,00,11,0
0,\
46,00,54,00,45,00,5f,00,45,00,72,00,72,00,6f,00,72,00,43,00,6f,00,75,00,6e,
\
00,74,00,4d,00,61,00,78,00,00,13,00,00,00,00,00,15,00,46,00,54,0
0,\
45,00,5f,00,50,00,72,00,6f,00,63,00,65,00,73,00,73,00,6f,00,72,00,41,00,66,
\
00,66,00,69,00,6e,00,69,00,74,00,79,00,00,15,00,ff,ff,ff,ff,ff,ff,ff,ff

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\chs]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisechs.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tschs.xml"
"Locale"=dword:00000804
"WBreakerClass"="{12CE94A0-DEFB-11D2-B31D-00600893A857}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\cht]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisecht.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tscht.xml"
"Locale"=dword:00000404
"WBreakerClass"="{1680E7C3-9430-4A51-9B82-1E7E7AEE5258}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\dan]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisedan.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsdan.xml"
"Locale"=dword:00000406
"WBreakerClass"="{16BC5CE4-2C78-4CB9-80D5-386A68CC2B2D}"
"StemmerClass"="{83BC7EF7-D27B-4950-A743-0F8E5CA928F8}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\deu]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisedeu.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsdeu.xml"
"Locale"=dword:00000407
"WBreakerClass"="{9EA69E16-AD5E-43ED-B90E-73D58771D3F6}"
"StemmerClass"="{6DE705A5-6467-43DC-9CE3-CCFE08B3F645}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\eng]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noiseeng.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tseng.xml"
"StemmerClass"="{D99F7670-7F1A-11CE-BE57-00AA0051FE20}"
"Locale"=dword:00000809
"WBreakerClass"="{173C97E2-AEBE-437C-9445-01B237ABF2F6}"

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\esn]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noiseesn.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsesn.xml"
"StemmerClass"="{B0516FF0-7F1C-11CE-BE57-00AA0051FE20}"
"Locale"=dword:00000c0a
"WBreakerClass"="{73D009FB-7D05-45AC-94C2-EDD1029942D4}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\fra]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisefra.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsfra.xml"
"StemmerClass"="{2A6EB050-7F1C-11CE-BE57-00AA0051FE20}"
"Locale"=dword:0000040c
"WBreakerClass"="{95679344-48B8-42CD-9B79-2F5784D70258}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\ita]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noiseita.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsita.xml"
"Locale"=dword:00000410
"WBreakerClass"="{FD86B5D0-12C6-11CE-BD31-00AA004BBB1F}"
"StemmerClass"="{6D36CE10-7F1C-11CE-BE57-00AA0051FE20}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\jpn]
"NoiseFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\noisejpn.txt"
"TsaurusFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\FTData\tsjpn.xml"
"Locale"=dword:00000411
"WBreakerClass"="{A9FFAE02-D834-47B3-8434-0D75DC0804BD}"
"StemmerClass"="{1305C72E-2073-452E-A72D-667032B3B4FE}"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSearch\Language\kor]

```



```
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisekor.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tskor.xml"
"Locale"=dword:00000412
"WBreakerClass"="{737FC9B5-BC66-4953-B7DB-B79A72A592A5}"
"StemmerClass"="{97E8EFC5-42DA-43ED-9CEC-84FDE3620F1A}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\Neutral]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noiseneu.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsglobal.xml"
"Locale"=dword:00000000
"WBreakerClass"="{C731071F-AC80-11D1-8DF3-00C04FB6EF4F}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\Ind]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisenid.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsnid.xml"
```

```
"Locale"=dword:00000413
"WBreakerClass"="{66B37110-8BF2-11CE-BE59-00AA0051FE20}"
"StemmerClass"="{860D28D0-8BF4-11CE-BE59-00AA0051FE20}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\plk]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisepk.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsplk.xml"
"Locale"=dword:00000415
"WBreakerClass"="{CA665B09-4642-4C84-A9B7-9B8F3CD7C3F6}"
"StemmerClass"="{B8713269-2D9D-4BF5-BF40-2615D75723D8}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\ptb]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisepb.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsptb.xml"
"Locale"=dword:00000416
"WBreakerClass"="{25B7FD48-5404-4BEB-9D80-B6982AF404FD}"
"StemmerClass"="{D5FCDD7E-DBFF-473F-BCCD-3AFD1890EA85}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\pts]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noiseps.txt"
```

```
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tspts.xml"
"WBreakerClass"="{5D5F3A69-620C-4952-B067-4D0126BB6086}"
"StemmerClass"="{D4171BC4-90BE-4F70-8610-DAB1C17F063C}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\rus]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noiserus.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsrus.xml"
"Locale"=dword:00000419
"WBreakerClass"="{20036404-F1AF-11D2-A57F-006052076F32}"
"StemmerClass"="{20036414-F1AF-11D2-A57F-006052076F32}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\sve]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisesve.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsve.xml"
"Locale"=dword:0000041d
```

```
"WBreakerClass"="{01C6B350-12C7-11CE-BD31-00AA004BBB1F}"
"StemmerClass"="{9478F640-7F1C-11CE-BE57-00AA0051FE20}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\tha]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisetha.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tstha.xml"
"Locale"=dword:0000041e
"WBreakerClass"="{CA22CF4-59FE-11D1-BBFF-00C04FB97FDA}"
"StemmerClass"="{CEDC01C7-59FE-11D1-BBFF-00C04FB97FDA}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\trk]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noisetrk.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tsrkr.xml"
"Locale"=dword:0000041f
"WBreakerClass"="{8DF412D1-62C7-4667-BBEC-38756576C21B}"
"StemmerClass"="{23A9C1C3-3C7A-4D2C-B894-4F286459DAD6}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\zh-hk]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noischt.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tscht.xml"
```

```
"WBreakerClass"="{1680E7C3-9430-4A51-9B82-1E7E7AEE5258}"
"StemmerClass"="{00000404}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\zh-mo]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noischt.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tscht.xml"
"Locale"=dword:00001404
"WBreakerClass"="{1680E7C3-9430-4A51-9B82-1E7E7AEE5258}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Language\\zh-sg]
"NoiseFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\noischts.txt"
"TaurusFile"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData\\tschts.xml"
"Locale"=dword:00001004
"WBreakerClass"="{12CE94A0-DEFB-11D2-B31D-00600893A857}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Plugins]
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSearch\\Plugins\\MicrosoftIndexer]
```

```
@="{68C6D24E-77E0-11D5-8528-00C04F68155C}"
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSQLServer]
"AuditLevel"=dword:00000002
"DebugClassID"="{38A0DD5B-F0AF-49F0-8B1B-828957E0764A}"
"DefaultLogin"="guest"
>LoginMode"=dword:00000002
"Map_#"=""
"Map#"=""
"Map$"=""
"SetHostName"=dword:00000000
"Tapeloadwaittime"=""
"BackupDirectory"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\Backup"
"SysDb"=dword:00000001
"FullTextDefaultPath"="C:\\Program Files\\Microsoft SQL
Server\\MSSQL.1\\MSSQL\\FTData"
"uptime_pid"=dword:00000234
"uptime_time_utc"=hex:60,0b,cb,f3,80,56,c5,01
```

```
[HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Microsoft SQL
Server\\MSSQL.1\\MSSQLServer\\CurrentVersion]
"Language"=dword:00000409
"CurrentVersion"="9.00.1116"
"RegisteredOwner"=""
```

```

"SerialNumber"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\Parameters]
"SQLArg0"="-dC:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\master.mdf"
"SQLArg1"="-eC:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\LOG\ERRORLOG"
"SQLArg2"="-iC:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\DATA\mastlog.ldf"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib]
"ForceEncryption"=dword:00000000
"HideInstance"=dword:00000000
"Certificate"=""
"DisplayName"="SQL Server Network Configuration"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\AdminConnection]
"DisplayName"="Dedicated Administrative Connection"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\AdminConnection\Tcp]
"TcpDynamicPorts"="1434"

"DisplayName"="TCP/IP"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Np]
"Enabled"=dword:00000001
"LocalAccessOnly"=dword:00000000
"PipeName"="\\.\pipe\sql\query"
"DisplayName"="Named Pipes"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Sm]
"Enabled"=dword:00000001
"DisplayName"="Shared Memory"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp]
"Enabled"=dword:00000001
"ListenOnAllIPs"=dword:00000001
"NoDelay"=dword:00000000
"KeepAlive"=dword:00007530
"DisplayName"="TCP/IP"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP1]
"Enabled"=dword:00000000
"Active"=dword:00000001

```

```

"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="15.1.101.78"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP2]
"Enabled"=dword:00000000
"Active"=dword:00000001
"TcpPort"="1433"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="10.193.25.78"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP3]
"Enabled"=dword:00000000
"Active"=dword:00000001
"TcpPort"="1433"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="127.0.0.1"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IPAll]
"TcpPort"="1433"

"TcpDynamicPorts"=""
"DisplayName"="Any IP Address"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Via]
"Enabled"=dword:00000001
"DefaultServerPort"="0:1433"
"ListenInfo"="1:1441[0x1],2:1442[0x2],3:1443[0x4],4:1444[0x8],5:1445[0x1
0],6:1446[0x20],7:1447[0x40],8:1448[0x80],9:1449[0x100],10:1450[0x200],
11:1451[0x400],12:1452[0x800],13:1453[0x1000],14:1454[0x2000],15:1455
[0x4000],16:1456[0x8000],16:1457[0x10000]"
"VendorName"="QLogic"
"DisplayName"="VIA"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\ADSDSOObject]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\DB2OLEDB]
"AllowInProcess"=dword:00000001

```

```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\OLEDB]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\MSDAORA]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\MSDASQL]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\MSIDXS]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\MSQLImpProv]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\MSSEARCHSQL]
"AllowInProcess"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\SQLNCLI]
"AllowInProcess"=dword:00000001
"DisallowAdhocAccess"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Providers\SQLOLEDB]
"AllowInProcess"=dword:00000001
"DisallowAdhocAccess"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Replication]
"IsInstalled"=dword:00000001
"DistributionDB"=""
"WorkingDirectory"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\REPLDATA"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\Setup]
"Version"="9.00.1116"
"Language"=dword:00000409
"Resume"=dword:00000000
"ProductCode"="{E83A5A6-0295-4EEB-8408-BCC919C13841}"
"SqlProgramDir"="C:\Program Files\Microsoft SQL Server\"
"checksum"=hex:38,38,33,66,66,66,30,65,62,38,32,62,30,34,37,62,31,30,35,
31,36,\

```

```

38,32,31,30,30,63,62,39,63,65,66,30,38,64,63,34,39,62,66,39,37,33,39,65,3
3,\
30,64,62,65,37,32,65,35,65,62,36,61,38,64,66,37,37,38,33,35,32,39,64,65,3
9,\
34,38,38,31,32,30,62,62,34,36,65,66,34,32,36,34,63,39,34,36,63,63,61,65,3
5,\
37,65,36,33,35,63,39,61,63,32,38,63,36,65,34,34,64,37,62,32,32,39,39,66,3
9,\
34,34,64,38,39,62,34,37,34,30,65,65,39,66,32,32,38,31,62,65,39,31,64,64,6
6,\
37,39,33,38,32,38,61,61,32,30,61,61,66,62,32,32,62,30,35,64,61,66,65,37,3
0,\
63,30,34,39,36,63,33,00
"Edition"="Beta Edition (64-bit)"
"SqlCluster"=dword:00000000
"FeatureList"="SQL_Engine=3 SQL_Data_Files=3 SQL_Replication=3
SQL_FullText=3"
"PatchLevel"="9.00.1116"
"Collation"="Latin1_General_BIN"
"FirstStart"=dword:00000000
"SQLDataRoot"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL"
"SQLPath"="C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL"

"SQLBinRoot"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\SQLServerAgent]
"JobHistoryMaxRowsPerJob"=dword:00000064
"SysAdminOnly"=dword:00000001
"DownloadedMaxRows"=dword:00000064
"ErrorLogFile"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\LOG\SQLAGENT.OUT"
"ErrorLoggingLevel"=dword:00000003
"JobHistoryMaxRows"=dword:000003e8
"MSXServerName"=""
"NonAlertableErrors"="1204,4002"
"RestartServer"=dword:00000001
"ServerHost"=""
"WorkingDirectory"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\JOBS\

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\SQLServerAgent\SubSystems]

```

```

"ANALYSISQUERY"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLOLAPSS90.DLL,NULL,OlapStart,OlapQue
ryEvent,100,100"
"ANALYSISQUERY"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLOLAPSS90.DLL,NULL,OlapStart,OlapCo
mmandEvent,OlapStop,100"
"ActiveScripting"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLATXSS90.DLL,NULL,ActiveScriptStart,Ac
tiveScriptEvent,ActiveScriptStop,10"
"CmdExec"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLCMDSS90.DLL,NULL,CmdExecStart,Cmd
Event,CmdExecStop,10"
"Distribution"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLREPS90.DLL,C:\Program
Files\Microsoft SQL
Server\90\COM\DISTRI.EXE,ReplStart,ReplEvent,ReplStop,100"
"Merge"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLREPS90.DLL,C:\Program
Files\Microsoft SQL
Server\90\COM\REPLMERG.EXE,ReplStart,ReplEvent,ReplStop,100"
"Snapshot"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\SQLREPS90.DLL,C:\Program
Files\Microsoft SQL
Server\90\COM\SNAPSHOT.EXE,ReplStart,ReplEvent,ReplStop,100"
"LogReader"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\sqlrepss90.dll,C:\Program Files\Microsoft
SQL Server\90\COM\logread.exe,ReplStart,ReplEvent,ReplStop,25"
"QueueReader"="C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn\sqlrepss90.dll,C:\Program Files\Microsoft
SQL Server\90\COM\qrdrvsc.exe,ReplStart,ReplEvent,ReplStop,100"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\SQLServerSCP]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\Analysis Server]
"Name"="MSSQLServerOLAPService"
"LNName"="MSOLAP$"
"Type"=dword:00000005

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\Full Text]
"LNName"="msftesql$"
"Name"="msftesql"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\Notification Services]
"LNName"="NS$"

```

```

"Type"=dword:00000008

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\Report Server]
"LNName"="ReportServer$"
"Name"="ReportServer"
"Type"=dword:00000006

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\ReportServer]
"Name"="ReportServer"
"LNName"="ReportServer$"
"Type"=dword:00000006

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\SQL Agent]
"Name"="SQLSERVERAGENT"
"LNName"="SQLAgent$"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\SQL Browser]
"LNName"=""
"Name"="SQLBrowser"
"Type"=dword:00000007

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\SQL Server]
"Name"="MSSQLServer"
"LNName"="MSSQL$"
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\Services\SSIS Server]

"LNName"=""
"Name"="MsDtsServer"
"Type"=dword:00000004

```

#### 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.

## 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,00,68,00,70,00,71,00,63,00,69,00,73,\
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\Parameters]
"CompletionMode"=dword:00000002
"EnableInterruptWatchdogTimer"=dword:00000001
"MinimumCosPollingInterval"=dword:00000005
"PerCpuDpCImportance"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\hpqcissb\Security]

"Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,0,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
00,00,02,00,88,00,06,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,0,\
05,12,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,00,\
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,04,00,00,0,0,\
00,14,00,8d,01,02,00,01,01,00,00,00,00,05,06,00,00,00,00,00,14,00,0,0,1,\
00,00,01,01,00,00,00,00,05,0b,00,00,00,00,18,00,fd,01,02,00,01,02,00,\
\
```

```
00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,05,12,00,00,0,\
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&0&2008"
"Count"=dword:0000003c
"NextInstance"=dword:0000003c
"1"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1457b150&0&0&2008"
"2"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&2b186594&0&0&2008"
"3"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&2b186594&0&0&2008"
"4"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3533f3f8&0&0&2008"
"5"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3533f3f3&0&0&2008"
"6"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&16ea0f70&0&0&2008"
"7"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&16ea0f70&0&0&2008"
"8"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1600be56&0&0&2008"
"9"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1600be56&0&0&2008"
"10"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&8a8e882&0&0&2008"
"11"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&8a8e882&0&0&2008"
"12"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&6aa0d5f&0&0&2008"
"13"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&6aa0d5f&0&0&2008"
"14"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3b7bf75a&0&0&2008"
"15"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3b7bf75a&0&0&2008"
"16"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&285861a&0&0&2008"
"17"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&285861a&0&0&2008"
"18"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&27eba306&0&0&2008"
"19"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&27eba306&0&0&2008"
"20"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&3b940db&4&0&0&2008"

```

```
"21"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&3b940db&4&0&0&2008"
"22"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&33c2a78d&0&0&2008"
"23"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&33c2a78d&0&0&2008"
"24"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&26156fbc&0&0&2008"
"25"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&26156fbc&0&0&2008"
"26"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1b9584b9&0&0&2008"
"27"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1b9584b9&0&0&2008"
"28"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&20dd8e1e&0&0&2008"
"29"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&20dd8e1e&0&0&2008"
"30"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&14635f0&0&0&2008"
"31"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&14635f0&0&0&2008"
"32"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13220302&0&0&2008"
"33"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13220302&0&0&2008"
"34"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1eb0d7f0&0&0&2008"
"35"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1eb0d7f0&0&0&2008"
"36"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&fd96073&0&0&2008"
"37"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&fd96073&0&0&2008"
"38"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&466fcd1&0&0&2008"
"39"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&466fcd1&0&0&2008"
"40"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&f78c9&0&0&2008"
"41"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&f78c9&0&0&2008"
"42"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&36ded50f&0&0&2008"
"43"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&36ded50f&0&0&2008"
"44"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&273093ea&0&0&2008"
"45"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&273093ea&0&0&2008"
"46"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&301d439f&0&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&348d8dd&0&2008"
"49"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&348d8dd&0&2808"
"50"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&214639d&8&0&2008"
"51"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&214639d&8&0&2808"
"52"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&1dded019&8&0&2008"
"53"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&1dded019&8&0&2808"
"54"="PCI\VEN_0E11&DEV_0046&SUBSYS_409C0E11&REV_01\6&13df1870&8&0&2008"
"55"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&13df1870&8&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&8&0&2008"
"59"="PCI\VEN_0E11&DEV_0046&SUBSYS_409D0E11&REV_01\6&19905e4a&8&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,00,32,00,33,00,30,00,30,00,30,\
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]

```

```

"MaximumSqlThreads"=dword:0000002b
"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&0&08"
"Count"=dword:00000010
"NextInstance"=dword:00000010
"1"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&251997c5&0&0&08"
"2"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&355cd0df&0&0&08"
"3"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&8f8ade6&0&0&08"
"4"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&345102a3&0&0&08"
"5"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&370af391&0&0&08"
"6"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&266723a4&0&0&08"
"7"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&23ad32b6&0&0&08"
"8"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&7ab2207&0&0&08"
"9"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&a6512f5&0&0&08"
"10"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&279b1ea1&0&0&08"
"11"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&22da925c&0&0&08"
"12"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&8331d07a6&0&0&08"
"13"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&b37b34f&0&0&08"
"14"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&3690080c&0&0&08"
"15"="PCI\VEN_1077&DEV_2312&SUBSYS_010C1077&REV_02\5&83949f8fa&0&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	0
allow updates	0	1	0	0	0
awe enabled	0	1	1	1	1
blocked process threshold	0	86400	0	0	0
c2 audit mode	0	1	0	0	0
clr enabled	0	1	0	0	0
cost threshold for parallelism	0	32767	5	5	5
cross db ownership chaining	0	1	0	0	0
cursor threshold	-1	2147483647	-1	-1	-1
Database Mail XPs	0	1	0	0	0
default full-text language	0	2147483647	1033	1033	1033
default language	0	9999	0	0	0
default trace enabled	0	1	0	0	0
disallow results from triggers	0	1	0	0	0
fill factor (%)	0	100	0	0	0
ft crawl bandwidth (max)	0	32767	100	100	100
ft crawl bandwidth (min)	0	32767	0	0	0
ft notify bandwidth (max)	0	32767	100	100	100
ft notify bandwidth (min)	0	32767	0	0	0
in-doubt xact resolution	0	2	0	0	0
index create memory (KB)	704	2147483647	0	0	0
lightweight pooling	0	1	1	1	1
locks	5000	2147483647	0	0	0
max degree of parallelism	0	64	1	1	1
max full-text crawl range	0	256	4	4	4
max server memory (MB)	16	2147483647	990000	990000	990000
max text repl size (B)	0	2147483647	65536	65536	65536
max worker threads	128	32767	2300	2300	2300
media retention	0	365	0	0	0
min memory per query (KB)	512	2147483647	1024	1024	1024
min server memory (MB)	0	2147483647	0	8	8
nested triggers	0	1	1	1	1
network packet size (B)	512	32767	4096	4096	4096
Ole Automation Procedures	0	1	0	0	0

open objects	0	2147483647	0	0
PH timeout (s)	1	3600	60	60
precompute rank	0	1	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
web 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: 05/31/05 15:03:00

System Name: \\DL1  
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Standard Edition

Version	5.2.3790 Service Pack 1 Build 3790
OS Manufacturer	Microsoft Corporation

```

System Manufacturer      HP1
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
(srv03_sp1_rtm.050324-1447)" Version = "5.2.3790.1830
User Name               Not Available
Time Zone               Pacific Daylight Time
Total Physical Memory   2,048.00 MB
Available Physical Memory 1.74 GB
Total Virtual Memory    3.85 GB
Available Virtual Memory 3.74 GB
Page File Space         2.00 GB
Page File               C:\pagefile.sys

```

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device	Status
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 0x00003B0-0x000003DF	PCI bus	
I/O Port 0x00003B0-0x000003DF	RAGE XL PCI Family (Microsoft Corporation)	

[DMA]

Resource	Device	Status
Channel 4	Direct memory access controller	OK

[Forced Hardware]

Device	PNP Device ID	Status
[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
0xFFFF80000-0xFFFFFFFF	System board	OK
0xA0000-0xBFFFF	PCI bus	OK

0xA0000-0xBFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0x80000000-0x80000000	PCI bus	OK
0xFC600000-0xFC6FFFFF	PCI bus	OK
0xFC900000-0xFE9FFFFF	PCI bus	OK
0xFD000000-0xFD0FFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xFE9FF000-0xFE9FFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xFE9FE000-0xFE9FEFFF	Standard OpenHCD USB Host Controller	OK
0xFC700000-0xFC7FFFFF	PCI bus	OK
0xFEAF0000-0xFEAFFFFF	PCI bus	OK
0xFEAF000-0xFEAFFFFF	QLogic QLA23xx PCI Fibre Channel Adapter	OK
0xFC800000-0xFC8FFFFF	PCI bus	OK
0xFEB00000-0xFEBFFFFF	PCI bus	OK
0xFEB00000-0xFEB0FFFF	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]

Can't Collect Information

[Video Codecs]

Can't Collect Information

[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 TypeNot Available
Adapter Description ATI Technologies Inc.
Adapter RAMNot Available
Installed Drivers Not Available
Driver Version Not Available
INF File 5.10.3663.6013 (atiixpad.inf section)
Color Planes ati2mpad
Color Table Entries Not Available
Resolution Not Available
Bits/Pixel Not Available
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]

Can't Collect Information

[Pointing Device]

Can't Collect Information

[Modem]

Item	Value
------	-------

[Network]

[Adapter]	Can't Collect Information [Printing]	afd	AFD Networking Support Environment c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes
Can't Collect Information	Name Driver Port Name Server Name	aha154x	aha154x	Running OK	Normal No
[Protocol]	b34-2525-b on red-prn-xrx (from V-JASGOE02) in session 0 Xerox Document Centre 255 PS TS001	aha154x	Not Available	Kernel Driver	No Disabled
Can't Collect Information	b35-4053-a on prn-corp1 (from V-JASGOE02) in session 0 Xerox Document Centre 255 PS TS002	Stopped	OK Normal	No	No
[WinSock]	[Problem Devices]	aic78u2	aic78u2	Not Available	Kernel Driver No Disabled
Can't Collect Information	Device PNP Device ID Error Code	Stopped	OK Normal	No	No
[USB]	[USB]	aic78xx	aic78xx	Not Available	Kernel Driver No Disabled
[Ports]	Device PNP Device ID	aliide	AliIde	Not Available	Kernel Driver No Disabled
[Serial]	Standard OpenHCD USB Host Controller PCI\VEN_1166&DEV_0221&SUBSYS_02201166&REV_05\3&267A616A&0&7A	alkernel	Altiris Kernel Driver c:\windows\system32\drivers\alkernel.sys	Kernel	Yes Normal
Can't Collect Information	USB Root Hub USB\ROOT_HUB\4&331E2EA4&0	Driver	Yes Manual Running OK	No	No
[Parallel]	[Software Environment]	asynmac	RAS Asynchronous Media Driver c:\windows\system32\drivers\asynmac.sys	Kernel	No Normal
Can't Collect Information	[System Drivers]	Driver	No Manual Stopped OK	No	No
[Storage]	Name Description File Type Started Start	atapi	Standard IDE/ESDI Hard Disk Controller c:\windows\system32\drivers\atapi.sys	Kernel Driver	Yes
[Drives]	Mode State Status Error Control Accept Pause Accept	atdisk	Boot Running OK Normal No	Yes	No
Can't Collect Information	abiosdsk Abiosdsk Not Available Kernel Driver No Disabled Stopped OK Ignore No	atdisk	Atdisk Not Available	Kernel Driver	No Disabled
[Disks]	acpi Microsoft ACPI Driver c:\windows\system32\drivers\acpi.sys	Stopped	OK Ignore No	No	No
Can't Collect Information	acpiec ACPIEC c:\windows\system32\drivers\acpiec.sys	Kernel	Yes	Normal	No
[SCSI]	adpu160m adpu160m Not Available Kernel Driver No Disabled Stopped OK Normal No	at2mpad	ati2mpad c:\windows\system32\drivers\ati2mpad.sys	Kernel Driver	Yes Manual Running OK
Can't Collect Information	adpu320 adpu320 Not Available Kernel Driver No Disabled Stopped OK Normal No	Driver	Ignore No Yes	No	No
[IDE]	afcnt afcnt Not Available Kernel Driver No Disabled Stopped OK Normal No	atmarpc	ATM ARP Client Protocol c:\windows\system32\drivers\atmarpc.sys	Kernel	No Normal
		Driver	No Manual Stopped OK	No	No
		audstub	Audio Stub Driver c:\windows\system32\drivers\audstub.sys	Kernel	Yes Normal
		Driver	Yes Manual Running OK	No	No
		b57w2k	Broadcom NetXtreme Gigabit Ethernet c:\windows\system32\drivers\b57xp32.sys	Kernel	Yes Normal
		Driver	Yes Manual Running OK	No	No
		beep	Beep c:\windows\system32\drivers\beep.sys	Kernel	Yes Normal
		Driver	Yes System Running OK	No	No
		cbidf2k	cbidf2k c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No Disabled Stopped OK
			Normal No No	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				
Driver	c:\windows\system32\drivers\clusdisk.sys		Kernel		Normal
	No	Disabled	Stopped	OK	
	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
cpqarray2	cpqarray2	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
crdisk	CRC Disk Filter Driver				
Driver	c:\windows\system32\drivers\crdisk.sys		Kernel		Normal
	Yes	Boot	Running	OK	
	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
	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	Boot	Running	OK	
dmload	dmload	c:\windows\system32\drivers\dmload.sys	Kernel Driver	Yes	Normal
	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
	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				
Driver	c:\windows\system32\drivers\i8042prt.sys		Kernel		Normal
	Yes	System	Running	OK	
	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				
Driver	c:\windows\system32\drivers\intelppm.sys		Kernel		Normal
	Yes	Manual	Running	OK	
	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				
Driver	c:\windows\system32\drivers\ipfltdrv.sys		Kernel		Normal
	No	Manual	Stopped	OK	
	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				
Driver	c:\windows\system32\drivers\ipsec.sys	Kernel	Normal		
	Yes	System	Running	OK	
	No	Yes			
ipsraidn	ipsraidn	Not Available	Kernel Driver	No	Disabled
	Stopped	OK	Normal	No	No
irenum	IR Enumerator Service				
Driver	c:\windows\system32\drivers\irenum.sys		Kernel		Normal
	No	Manual	Stopped	OK	
	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				
Driver	c:\windows\system32\drivers\kbdclass.sys		Kernel		Normal
	Yes	System	Running	OK	
	No	Yes			
ksecdd	KSecDD	c:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Running
	Normal	No	Yes		OK
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
	Kernel Driver	Yes	System	Running	OK
	Ignore	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
rasppoe	Remote Access PPPOE Driver c:\windows\system32\drivers\rasppoe.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
raspti	Direct Parallel c:\windows\system32\drivers\raspti.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
rdbss	Rdbss c:\windows\system32\drivers\rdbss.sys	File System Driver	Yes	System	Running	OK	Normal	Yes
rdpcdd	RDPCDD c:\windows\system32\drivers\rdpcdd.sys	Kernel Driver	Yes	System	Running	OK	Ignore	Yes
rdpdr	Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
rdpwd	RDPWD c:\windows\system32\drivers\rdpwd.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	No
secdrv	Secdrv c:\windows\system32\drivers\secdrv.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
serenum	Serenum Filter Driver c:\windows\system32\drivers\serenum.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
serial	Serial port driver c:\windows\system32\drivers\serial.sys	Kernel Driver	No	System	Stopped	OK	Ignore	No
sfloppy	Sfloppy c:\windows\system32\drivers\sfloppy.sys	Kernel Driver	No	System	Stopped	OK	Ignore	No
simbad	Simbad Not Available	Kernel Driver	No	Normal	Stopped	OK	Normal	No
sparrow	Sparrow Not Available	Kernel Driver	No	Normal	Stopped	OK	Normal	No

srv	Srv System Driver	c:\windows\system32\drivers\svr.sys	File	Yes	Manual	Running	OK	Normal
swenum	Software Bus Driver	c:\windows\system32\drivers\swenum.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
symc810	symc810	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
symc8xx	symc8xx	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
symmpi	symmpi	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
sym_hi	sym_hi	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
sym_u3	sym_u3	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
tcpip	TCP/IP Protocol Driver c:\windows\system32\drivers\tcpip.sys	Kernel Driver	Yes	System	Running	OK	Normal	No
tdpipe	TDPIPE c:\windows\system32\drivers\tdpipe.sys	Kernel Driver	No	Manual	Stopped	OK	Ignore	No
tdtcp	TDTCP c:\windows\system32\drivers\tdtcp.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	No
termdd	Terminal Device Driver c:\windows\system32\drivers\termdd.sys	Kernel Driver	Yes	System	Running	OK	Normal	No
toside	TosIde Not Available	Kernel Driver	No	Normal	Stopped	OK	Normal	No
udfs	Udfs System Driver	c:\windows\system32\drivers\udfs.sys	File	No	Disabled	Stopped	OK	Normal
ultra	ultra	Not Available	Kernel Driver	No	Stopped	OK	Normal	No
update	Microcode Update Driver c:\windows\system32\drivers\update.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No

usbhub	USB2 Enabled Hub c:\windows\system32\drivers\usbhub.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
usbohci	Microsoft USB Open Host Controller Miniport Driver c:\windows\system32\drivers\usbohci.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
usbstor	USB Mass Storage Driver c:\windows\system32\drivers\usbstor.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
vgasave	VGA Display Controller. c:\windows\system32\drivers\vga.sys	Kernel Driver	Yes	System	Running	OK	Ignore	No
viaide	ViaIde Not Available	Kernel Driver	No	Stopped	OK	Normal	Normal	No
volsnap	Storage volumes c:\windows\system32\drivers\volsnap.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
wanarp	Remote Access IP ARP Driver c:\windows\system32\drivers\wanarp.sys	Kernel Driver	Yes	Manual	Running	OK	Normal	No
wdica	WDICA Not Available	Kernel Driver	No	Stopped	OK	Ignore	Normal	No
wlbs	Network Load Balancing c:\windows\system32\drivers\wlbs.sys	Kernel Driver	No	Manual	Stopped	OK	Normal	No
x86perfsys	Low Overhead Profiler Kernel Driver Ignore	Kernel Driver	No	Manual	Stopped	OK	Normal	No

[Signed Drivers]

Device Name	Signed Date	Manufacturer	Device Class	INF Name	Driver Name	Version	Device ID	Driver
Microsoft System Management	5.2.3790.1830		BIOS	ROOT\SYSTEM\0002	Yes	10/1/2002	(Standard system devices)	SYSTEM
Microcode Update Device	10/1/2002		Yes	ROOT\SYSTEM\0001	SYSTEM	5.2.3790.0	(Standard system devices)	Not Available
Plug and Play Software Device Enumerator	5.2.3790.0		10/1/2002	ROOT\SYSTEM\0000	Yes	10/1/2002	(Standard system devices)	SYSTEM

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)		
machine.inf	Not Available	ROOT\RDPDR\0000		
Direct Parallel	Yes	NET	5.2.3790.0	
10/1/2002	Microsoft	netrasa.inf	Not Available	
ROOT\MS_PTMINIIMPORT\0000				
WAN Miniport (PPTP)	Yes	NET	5.2.3790.0	
10/1/2002	Microsoft	netrasa.inf	Not Available	
ROOT\MS_PPTPMINIIMPORT\0000				
WAN Miniport (PPPOE)	Yes	NET	5.2.3790.0	
10/1/2002	Microsoft	netrasa.inf	Not Available	
ROOT\MS_PPPOEMINIIMPORT\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_L2TPMINIIMPORT\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
Available	ROOT\MEDIA\MS_MMMCI			
Legacy Audio Drivers	Yes	MEDIA	5.2.3790.0	
10/1/2002	(Standard system devices)	wave.inf		Not Available
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
Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_X86PERFSYS\0000				
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER		
Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_WANARP\0000				

volsnap	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
VGA Display	Root\LEGACY_VGA\0000	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_VGASAVE\0000					
TDTCP	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_TDTCP\0000					
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_TCP\0000					
sacdrv	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_SACDRV\0000					
RDPWD	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_RDPWD\0000					
RDPDCC	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_RDPDCC\0000					
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER			
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_RASACD\0000					
qlvika	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_QLV\0000					
Page Tracker1 for X86perfsys	Not Available	LEGACYDRIVER			
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_PGTRACKR01\0000					
Partition Manager	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_PARTMGR\0000					
Null	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NULL\0000					
Network Monitor Driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NM\0000					
NetBios over Tcpip	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NETBT\0000					
NDProxy	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NDPROXY\0000					
NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER			
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NDISUIO\0000					
Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER			
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_NDIS\0000					

NDIS System Driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
mountmgr	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_MOUNTMGR\0000					
modem	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_MODEM\0000					
mnmdd	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_MNMDD\0000					
ksecdd	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_KSECDD\0000					
IPSEC driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_IPSEC\0000					
HTTP	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_HTTP\0000					
Generic Packet Classifier	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_GPC\0000					
Fips	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_FIPS\0000					
dmload	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_DMLoad\0000					
dmbboot	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_DMBOOT\0000					
CRC Disk Filter Driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_CRCDISK\0000					
Beep	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_BEEP\0000					
Altiris Kernel Driver	Not Available	LEGACYDRIVER			Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
ROOT\LEGACY_ALKERNEL\0000					
AFD Networking Support Environment	Not Available	LEGACYDRIVER			Not Available
Available	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&SIGNATUREB107B107OFF					
SET7E0LENGTH12A1C90400					
Volume Manager	Yes	SYSTEM	5.2.3790.0		
10/1/2002	(Standard system devices)	machine.inf		Not Available	
Available	ROOT\FDISK\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&107  
 0020&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&107  
 0020&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&13C  
 0B0C5&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&267  
 A616A&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&267  
 A616A&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&267  
 A616A&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&267  
 A616A&0&7E  
 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  
 \_\_\_\_\_4A3544564745343920202020202020202020202020202020  
 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  
 PCI\VEN\_1166&DEV\_0203&SUBSYS\_00000000&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\AV00402\4&89B5141&0&80000001&00&03  
 RAGE XL PCI Family (Microsoft Corporation) Yes DISPLAY  
 5.10.2600.6014 8/8/2001 ATI Technologies Inc.  
 atixpad.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\_00000000&REV\_00\3&267  
 A616A&0&01  
 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\_00000000&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  
 Available Not Available Not Available HTREE\ROOT\0  
 Not Available Yes Not Available 2:5:2 Not Available Not Available  
 Available Not Available Not Available b34-2525-b on red-pm-xrx (from V-JASGOE02) in session 0

Not Available Yes Not Available 2:5.2 Not Available Not Available Not Available b35-4053-a on prn-corp1 (from V-JASGOE02) in session 0

[Environment Variables]

Variable	Value	User Name
ComSpec	%SystemRoot%\system32\cmd.exe	<SYSTEM>
Path	%SystemRoot%\system32;%SystemRoot%\System32\Wbem;C:\Program Files\Microsoft SQL Server\80\Tools\BINN<SYSTEM>	
windir	%SystemRoot%\Windows_NT	<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>
FP_NO_HOST_CHECK	NO	<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	
TEMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE	
TEMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE	
TEMP	%USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE	
TEMP	%USERPROFILE%\Local Settings\Temp DL1\Administrator	
TEMP	%USERPROFILE%\Local Settings\Temp DL1\Administrator	

[Print Jobs]

Document	Size	Owner	Notify	Status	Time
Submitted	Start Time	Until Time	Elapsed Time	Pages Printed	
Processor	Job ID	Priority	Parameters	Driver	Print
	Host Print Queue		Data Type	Name	

[Network Connections]

Local Name	Remote Name	Type	Status	User	
[Running Tasks]					
Name	Path	Process ID	Priority	Min Working Set	
Max Working Set	File Date	Start Time	Version	Size	
system idle process		Not Available	0	0	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available
system	Not Available	4	8	0	1413120
Available	Not Available	Not Available	Not Available	Not Available	Not Available
smss.exe	Not Available	572	11	204800	1413120
Available	5/28/2005 12:33 AM		Not Available	Not Available	Not Available
csrss.exe	c:\windows\system32\csrss.exe		636	13	
Available	204800	1413120	5/28/2005 12:33 AM		
winlogon.exe	c:\windows\system32\winlogon.exe		676	13	
Available	204800	1413120	5/28/2005 12:33 AM		
services.exe	c:\windows\system32\services.exe		720	9	
Available	204800	1413120	5/28/2005 12:33 AM		
lsass.exe	c:\windows\system32\lsass.exe		748	9	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		952	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1016	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1124	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1184	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1216	8	
Available	204800	1413120	5/28/2005 12:33 AM		

spoolsv.exe	c:\windows\system32\spoolsv.exe		1720	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1752	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		2000	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		376	8	
Available	204800	1413120	5/28/2005 12:33 AM		
rsys.exe	c:\program files\benchcraft\rsys.exe		408	8	
Available	204800	1413120	5/28/2005 12:33 AM		
svchost.exe	c:\windows\system32\svchost.exe		1472	8	
Available	204800	1413120	5/28/2005 12:33 AM		
wmiprvse.exe	c:\windows\system32\wbem\wmiprvse.exe		220	8	
Available	204800	1413120	5/28/2005 12:35 AM		
dllhost.exe	c:\windows\system32\dllhost.exe		876	8	
Available	204800	1413120	5/28/2005 4:21 AM		
csrss.exe	c:\windows\system32\csrss.exe		3772	13	
Available	204800	1413120	5/28/2005 2:36 PM		
winlogon.exe	c:\windows\system32\winlogon.exe		4320	13	
Available	204800	1413120	5/28/2005 2:36 PM		
rdpclip.exe	c:\windows\system32\rdpclip.exe		7796	8	
Available	204800	1413120	5/28/2005 2:36 PM		
explorer.exe	c:\windows\explorer.exe		8872	8	204800
Available	204800	1413120	5/28/2005 2:36 PM		

```

acntusr.exe c:\program files\altiris\acient\acntusr.exe 4072
8 204800 1413120 5/28/2005 2:36 PM
6, 1, 375 180.00 KB (184,320 bytes) 5/3/2005 8:59 AM

taskmgr.exe c:\windows\system32\taskmgr.exe 8816 13
204800 1413120 5/28/2005 2:36 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 164.50
KB (168,448 bytes) 4/14/2005 10:01 AM

logon.scr c:\windows\system32\logon.scr 780 4
204800 1413120 5/28/2005 2:46 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50
KB (509,440 bytes) 4/14/2005 10:01 AM

inetinfo.exe c:\windows\system32\inetrv\inetinfo.exe 1776
8 204800 1413120 5/31/2005 11:22 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 4536 8
204800 1413120 5/31/2005 11:22 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
6544 8 204800 1413120
5/31/2005 3:01 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
"c:\windows\system32\csrss.exe"				"c:\windows\system32\csrss.exe"
"c:\windows\system32\ntdll.dll"				"c:\windows\system32\ntdll.dll"
"c:\windows\system32\csrssv.dll"				"c:\windows\system32\csrssv.dll"
"c:\windows\system32\basesrv.dll"				"c:\windows\system32\basesrv.dll"
"c:\windows\system32\winsrv.dll"				"c:\windows\system32\winsrv.dll"
"c:\windows\system32\gdi32.dll"				"c:\windows\system32\gdi32.dll"
"c:\windows\system32\advapi32.dll"				"c:\windows\system32\advapi32.dll"
"c:\windows\system32\kernel32.dll"				"c:\windows\system32\kernel32.dll"
"c:\windows\system32\prcrt4.dll"				"c:\windows\system32\prcrt4.dll"
"c:\windows\system32\user32.dll"				"c:\windows\system32\user32.dll"
"c:\windows\system32\sxs.dll"				"c:\windows\system32\sxs.dll"

```

"c:\windows\system32\winlogon.exe"
"c:\windows\system32\cryptsp.dll"
"c:\windows\system32\crypt32.dll"
"c:\windows\system32\msasn1.dll"
"c:\windows\system32\msasn1.dll"
"c:\windows\system32\msvcrt.dll"
"c:\windows\system32\msvcrt.dll"
"c:\windows\system32\nddeapi.dll"
"c:\windows\system32\nddeapi.dll"
"c:\windows\system32\profmap.dll"
"c:\windows\system32\profmap.dll"
"c:\windows\system32\netapi32.dll"
"c:\windows\system32\netapi32.dll"
"c:\windows\system32\userenv.dll"
"c:\windows\system32\userenv.dll"
"c:\windows\system32\psapi.dll"
"c:\windows\system32\psapi.dll"
"c:\windows\system32\regapi.dll"
"c:\windows\system32\regapi.dll"
"c:\windows\system32\secur32.dll"
"c:\windows\system32\secur32.dll"
"c:\windows\system32\setupapi.dll"
"c:\windows\system32\setupapi.dll"
"c:\windows\system32\version.dll"
"c:\windows\system32\version.dll"
"c:\windows\system32\winsta.dll"
"c:\windows\system32\winsta.dll"
"c:\windows\system32\ws2_32.dll"
"c:\windows\system32\ws2_32.dll"
"c:\windows\system32\ws2help.dll"
"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
ole32 6.0.3790.1830 (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

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

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

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

winotify 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\winotify.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

adslidpc 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\adslidpc.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

scredir 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 28.00 KB (28,672 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\scredir.dll

cscui 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 319.50 KB (327,168 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\cscui.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

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

rdpsnd 5.2.3790.0 (srv03\_rtm.030324-2048) 18.00 KB (18,432 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\rdpsnd.dll

printui 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 563.00 KB (576,512 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\printui.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

msacm32 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 22.00 KB (22,528 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msacm32.drv

msacm32 5.2.3790.1830 (srv03\_sp1\_rtm.050324-1447) 69.50 KB (71,168 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msacm32.dll

imaadp32 5.2.3790.0 (srv03\_rtm.030324-2048) 15.50 KB (15,872 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\imaadp32.acm

msadp32 5.2.3790.0 (srv03\_rtm.030324-2048) 14.50 KB (14,848 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msadp32.acm

msg711 5.2.3790.0 (srv03\_rtm.030324-2048) 10.00 KB (10,240 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msg711.acm

msgsm32 5.2.3790.0 (srv03\_rtm.030324-2048) 20.50 KB (20,992 bytes) 4/12/2005 1:42 PM Microsoft Corporation c:\windows\system32\msgsm32.acm

tssoft32 1.01 9.50 KB (9,728 bytes) 4/12/2005 1:43 PM DSP GROUP, INC. c:\windows\system32\tssoft32.acm

tsd32 1.03 16.50 KB (16,896 bytes) 4/12/2005 1:43 PM DSP GROUP, INC. c:\windows\system32\tsd32.dll

msg723 5.2.3790.1830 120.00 KB (122,880 bytes) 4/14/2005 10:01 AM Microsoft Corporation c:\windows\system32\msg723.acm

msaud32 8.00.00.4487288.00 KB (294,912 bytes) 4/12/2005 1:43 PM Microsoft Corporation c:\windows\system32\msaud32.acm

sl\_anet 3.02 84.00 KB (86,016 bytes) 4/12/2005 1:43 PM Sipro Lab Telecom Inc. c:\windows\system32\sl\_anet.acm

l3codeca 1, 9, 0, 0305 284.00 KB (290,816 bytes) 4/12/2005 1:43 PM Fraunhofer Institut Integrierte Schaltungen IIS c:\windows\system32\l3codeca.acm

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

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

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

scsvr 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\scsvr.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

umpnprmg 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\umpnprmg.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				
scedi	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\scedi.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				
dnssrslvr	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\dnssrslvr.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				
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				
winnr	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\winnr.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_6595b64144cc1df_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			
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			
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			
trkws	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\trkws.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			
wuauerv	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\wuauerv.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			
mspatcha	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	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			
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			
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			
rasdlg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	663.00 KB (678,912 bytes)	4/12/2005 1:42 PM	Microsoft Corporation
	c:\windows\system32\rasdlg.dll			

xactsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	90.00 KB (92,160 bytes)	4/14/2005 10:00 AM	Microsoft Corporation
	c:\windows\system32\xactsrv.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 (dnsv3(skatar).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			

psui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	135.00 KB (138,240 bytes)	4/14/2005 10:26 AM	Microsoft Corporation
	c:\windows\system32\pool\drivers\w32x86\3\psui.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	5.31.23.1224439.00 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	
	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
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			
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			
rdpclip	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	68.00 KB (69,632 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\rdpclip.exe			
urlmon	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	673.00 KB (689,152 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\urlmon.dll			
explorer	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.00 MB (1,050,624 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\explorer.exe			
browseui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1,009.00 KB (1,033,216 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\browseui.dll			
shdocvw	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.43 MB (1,502,720 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\shdocvw.dll			
themeui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	377.50 KB (386,560 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\themeui.dll			

msimg32	5.2.3790.0 (srv03_rtm.030324-2048)	4.50 KB (4,608 bytes)	4/12/2005 1:42 PM	Microsoft Corporation
actxprxy	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	96.50 KB (98,816 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\actxprxy.dll			
linkinfo	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	19.00 KB (19,456 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\linkinfo.dll			
ntshrui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	140.00 KB (143,360 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\ntshrui.dll			
webcheck	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	272.50 KB (279,040 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\webcheck.dll			
stobject	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\stobject.dll			
batmeter	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	31.50 KB (32,256 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\batmeter.dll			
powrprof	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	16.50 KB (16,896 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\powrprof.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			
adintusr	6, 1, 375	180.00 KB (184,320 bytes)	5/3/2005 8:59 AM	
	c:\program files\altiris\aclient\adintusr.exe			
taskmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	164.50 KB (168,448 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\taskmgr.exe			
vdmdbg	5.2.3790.0 (srv03_rtm.030324-2048)	25.00 KB (25,600 bytes)	4/12/2005 1:43 PM	Microsoft Corporation
	c:\windows\system32\vdmdbg.dll			
util.dll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.50 KB (29,184 bytes)	4/14/2005 10:01 AM	Microsoft Corporation
	c:\windows\system32\util.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
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\inetrv\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\inetrv\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\inetrv\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\inetrv\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\inetrv\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\inetrv\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\inetrv\metadata.dll
svcxext 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\inetrv\svcxext.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\inetrv\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\inetrv\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\inetrv\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\inetrv\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\inetrv\lonsint.dll
[Services]
Display Name Name State Start Mode Service
Type Path Error Control Start Name Tag ID
[Program Groups]
Can't Collect Information
[Startup Programs]
Can't Collect Information
[OLE Registration]
Can't Collect Information
[Windows Error Reporting]
Time Type Details
[Internet Settings]
[Internet Explorer]
[ Following are sub-categories of this main category ]
[Summary]
Can't Collect Information
[File Versions]
Can't Collect Information
[Connectivity]
Can't Collect Information
[Cache]

```

[Summary] are sub-categories of this main category ]

Can't Collect Information

[List of Objects]

Can't Collect Information

[Content]

[ Following are sub-categories of this main category ]  
[Summary]

Can't Collect Information

[Personal Certificates]

Can't Collect Information

[Other People Certificates]

Can't Collect Information

[Publishers]

Can't Collect Information

[Security]

Can't Collect Information

### COM+ Settings

TPCC.AITxns:

Activation:

Enable Object Pooling selected  
Minimum Pool Size: 54  
Maximum Pool Size: 54

Creation Timeout: 60000  
 Enable Object Construction  
 Enable Just In Time Activation  
 Concurrency:  
 Concurrency Required

"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,\\

"First Help"=dword:0000005204  
 "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

**Microsoft IIS Registry Parameters**

Windows Registry Editor Version 5.00

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]  
 "ListenBackLog"=dword:00008ca0  
 "PoolThreadLimit"=dword:00000708  
 "MaxPoolThreads"=dword:00000ffe  
 "ThreadTimeout"=dword:00015180

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):52,00,50,00,43,00,53,00,53,00,00,00,48,00,54,0,54,00,\\

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]  
 "Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,0,0,2,\\

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]  
 "Library"="infoctrs.dll"  
 "Open"="OpenINFOPerformanceData"  
 "Close"="CloseINFOPerformanceData"  
 "Collect"="CollectINFOPerformanceData"  
 "Last Counter"=dword:00000c50  
 "Last Help"=dword:00000c51  
 "First Counter"=dword:00000c10  
 "First Help"=dword:00000c11  
 "Library Validation Code"=hex:59,fa,4c,b9,fc,d3,c4,01,10,25,00,00,00,00,00,00  
 "WbemAdapFileTime"=hex:00,a0,99,79,23,37,c3,01  
 "WbemAdapFileSize"=dword:00002510  
 "WbemAdapStatus"=dword:00000000

50,00,46,00,69,00,6c,00,74,00,65,00,72,00,00,00,49,00,49,00,53,00,41,00,44,\\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:80,51,01,00,00,00,00,00,00,00,00,03,00,00,00,14,0,0,\\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\\00,00,02,00,88,00,06,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\\

**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

00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,0  
 [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,00,57,00,53,\\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,00,69,00,69,00,73,00,77,00,33,00,61,00,64,\\00,6d,00,2e,00,64,00,6c,00,6c,00,00,00  
 "AcceptExOutstanding"=dword:00000028

05,12,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,00,\\20,02,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,04,00,00,00,0,\\00,14,00,8d,01,02,00,01,01,00,00,00,00,05,06,00,00,00,00,14,00,00,1,\\00,00,01,01,00,00,00,00,05,0b,00,00,00,00,18,00,fd,01,02,00,01,02,00,0,\\00,00,00,05,20,00,00,23,02,00,00,01,01,00,00,00,00,05,12,00,00,0,\\01,01,00,00,00,00,05,12,00,00,00  
 [HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]  
 "0"="Root\\LEGACY\_W3SVC\\00000"  
 "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/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:000008c0
"MaxPendingDeliveries"=dword:000009c4
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"

```

```

"DbServer"="via:sqlorca"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="ssdl"
"COM_SinglePool"="YES"
"sprefix"="sp0"
"ConnectDelay"=dword:000003e8
"CallNoDuplicatesNewOrder"=dword:00000000

```

NOTE: This is representative of 1 web client. Sprefix was varied from 0-15 by every 2 web clients.

### TCPIP Registry Parameters

Windows Registry Editor Version 5.00

```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"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:0000ffff
"DhcpNameServer"="15.1.101.1"
"DhcpDomain"="hp-perf.net"
```

### Client VIA Configuration

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\Client\SuperSocketNetLib\VIA]
"Vendor"="QLogic"
"DefaultServerPort"="1:1434"
"DefaultClientNIC"="0"
"RecognizedVendors"="Giganet, ServerNet II,QLogic"
NOTE: DefaultServerPort was varied on every two web clients to connect to the appropriate SoftNuma node. In addition web client #32 used a separate connection port to access the last 17th Numa Node of 1 processor (See System Overview for description)
```

## RTE Input Parameters

### 3Tier.pro

```
Profile: 3-Tier-91991WH-Dist-1
File Path: D:\YukonTests\BenchCraft Profiles\VIA\3-Tier\3-Tier-91991WH-Dist-1.xml
Version: 5

Number of Engines: 64
```

```
Name: DRIVER.1.01
Description:
Directory: C:\DRIVER.1.1.log
Machine: CL1
Parameter Set: Audit
Index: 0
Seed: 73660
Configured Users: 14330
Pipe Name: DRIVER.1.1
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp0
```

```
Name: DRIVER.1.02
Description:
Directory: C:\DRIVER.1.2.log
```

```
Machine: CL1
Parameter Set: Audit
Index: 25000000
Seed: 73660
Configured Users: 14340
Pipe Name: DRIVER.1.2
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp1
```

```
Name: DRIVER.1.03
Description:
Directory: C:\DRIVER.1.3.log
Machine: CL2
Parameter Set: Audit
Index: 50000000
Seed: 73660
Configured Users: 14340
Pipe Name: DRIVER.1.3
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp2
```

```
Name: DRIVER.1.04
```

```
Description: C:\DRIVER.1.4.log
Machine: CL2
Parameter Set: Audit
Index: 75000000
Seed: 73660
Configured Users: 14340
Pipe Name: DRIVER.1.4
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp3
```

```
Name: DRIVER.1.05
Description:
Directory: C:\DRIVER.1.5.log
Machine: CL3
```

```
Parameter Set: Audit
Index: 100000000
Seed: 73660
Configured Users: 14480
Pipe Name: DRIVER.1.5
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: spprefix=sp4
```

```
Name: DRIVER.1.06
Description:
Directory: C:\DRIVER.1.6.log
Machine: CL3
Parameter Set: Audit
Index: 125000000
Seed: 73660
Configured Users: 14480
Pipe Name: DRIVER.1.6
Connect Rate: 0
Start Rate: 0
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: spprefix=sp5
```

```
Name: DRIVER.1.07
Description:
```

Directory: C:\DRIVER.1.7.log  
Machine: CL4  
Parameter Set: Audit  
Index: 150000000  
Seed: 73660  
Configured Users: 14480  
Pipe Name: DRIVER.1.7  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp6

Name: DRIVER.1.08  
Description:  
Directory: C:\DRIVER.1.8.log  
Machine: CL4  
Parameter Set: Audit  
Index: 175000000  
Seed: 73660  
Configured Users: 14480  
Pipe Name: DRIVER.1.8  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp7

Name: DRIVER.1.09  
Description:  
Directory: C:\DRIVER.1.9.log  
Machine: CL5  
Parameter Set: Audit  
Index: 200000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.9  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp8

Name: DRIVER.1.10  
Description:  
Directory: C:\DRIVER.1.10.log

Parameter Set: Audit  
Index: 225000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.10  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp9

Name: DRIVER.1.11  
Description:  
Directory: C:\DRIVER.1.11.log  
Machine: CL6  
Parameter Set: Audit  
Index: 250000000  
Seed: 73660  
Configured Users: 14630

Pipe Name: DRIVER.1.11  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp10

Name: DRIVER.1.12  
Description:  
Directory: C:\DRIVER.1.12.log  
Machine: CL6  
Parameter Set: Audit  
Index: 275000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.12  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp11

Name: DRIVER.1.13  
Description:  
Directory: C:\DRIVER.1.13.log  
Machine: CL7

Parameter Set: Audit  
Index: 300000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.13  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp12

Name: DRIVER.1.14  
Description:  
Directory: C:\DRIVER.1.14.log  
Machine: CL7  
Parameter Set: Audit  
Index: 325000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.14

Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp13

Name: DRIVER.1.15  
Description:  
Directory: C:\DRIVER.1.15.log  
Machine: CL8  
Parameter Set: Audit  
Index: 350000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.15  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp14

Name: DRIVER.1.16  
Description:  
Directory: C:\DRIVER.1.16.log  
Machine: CL8  
Parameter Set: Audit



Index: 37500000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.16  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp15

Name: DRIVER.1.17  
Description:  
Directory: C:\DRIVER.1.17.log  
Machine: CL9  
Parameter Set: Audit  
Index: 40000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.17  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp16

Name: DRIVER.1.18  
Description:  
Directory: C:\DRIVER.1.18.log  
Machine: CL9  
Parameter Set: Audit  
Index: 425000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.18  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp17

Name: DRIVER.1.19  
Description:  
Directory: C:\DRIVER.1.19.log  
Machine: CL10  
Parameter Set: Audit  
Index: 450000000

Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.19  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp18

Name: DRIVER.1.20  
Description:  
Directory: C:\DRIVER.1.20.log  
Machine: CL10  
Parameter Set: Audit  
Index: 475000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.20  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1

Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp19

Name: DRIVER.1.21  
Description:  
Directory: C:\DRIVER.1.21.log  
Machine: CL11  
Parameter Set: Audit  
Index: 500000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.21  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp20

Name: DRIVER.1.22  
Description:  
Directory: C:\DRIVER.1.22.log  
Machine: CL11  
Parameter Set: Audit  
Index: 525000000  
Seed: 73660

Pipe Name: DRIVER.1.23  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp21

Name: DRIVER.1.23  
Description:  
Directory: C:\DRIVER.1.23.log  
Machine: CL12  
Parameter Set: Audit  
Index: 550000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.23  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10

CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp22

Name: DRIVER.1.24  
Description:  
Directory: C:\DRIVER.1.24.log  
Machine: CL12  
Parameter Set: Audit  
Index: 575000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.24  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp23

Name: DRIVER.1.25  
Description:  
Directory: C:\DRIVER.1.25.log  
Machine: CL13  
Parameter Set: Audit  
Index: 600000000  
Seed: 73660  
Configured Users: 14630

Pipe Name: DRIVER.1.25  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp24

Name: DRIVER.1.26  
Description:  
Directory: C:\DRIVER.1.26.log  
Machine: CL13  
Parameter Set: Audit  
Index: 625000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.26  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp25

Name: DRIVER.1.27  
Description:  
Directory: C:\DRIVER.1.27.log  
Machine: CL14  
Parameter Set: Audit  
Index: 650000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.27  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp26

Name: DRIVER.1.28  
Description:  
Directory: C:\DRIVER.1.28.log  
Machine: CL14  
Parameter Set: Audit  
Index: 675000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.28

Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp27

Name: DRIVER.1.29  
Description:  
Directory: C:\DRIVER.1.29.log  
Machine: CL15  
Parameter Set: Audit  
Index: 700000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.29  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0

Additional Options: spprefix=sp28

Name: DRIVER.1.30  
Description:  
Directory: C:\DRIVER.1.30.log  
Machine: CL15  
Parameter Set: Audit  
Index: 725000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.30  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp29

Name: DRIVER.1.31  
Description:  
Directory: C:\DRIVER.1.31.log  
Machine: CL16  
Parameter Set: Audit  
Index: 750000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.31  
Connect Rate: 0

Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp30

Name: DRIVER.1.32  
Description:  
Directory: C:\DRIVER.1.32.log  
Machine: CL16  
Parameter Set: Audit  
Index: 775000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.32  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp31

Name: DRIVER.1.33  
Description:  
Directory: C:\DRIVER.1.33.log  
Machine: CL17  
Parameter Set: Audit  
Index: 800000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.33  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp32

Name: DRIVER.1.34  
Description:  
Directory: C:\DRIVER.1.34.log  
Machine: CL17  
Parameter Set: Audit  
Index: 825000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.34  
Connect Rate: 0  
Start Rate: 0

Max. Concurrency: -1  
Concurrency Rate: 20  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp33

Name: DRIVER.1.35  
Description:  
Directory: C:\DRIVER.1.35.log  
Machine: CL18  
Parameter Set: Audit  
Index: 850000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.35  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp34

Name: DRIVER.1.36  
Description:  
Directory: C:\DRIVER.1.36.log  
Machine: CL18  
Parameter Set: Audit  
Index: 875000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.36  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp35

Name: DRIVER.1.37  
Description:  
Directory: C:\DRIVER.1.37.log  
Machine: CL19  
Parameter Set: Audit  
Index: 900000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.37  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1

CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp36

Name: DRIVER.1.38  
Description:  
Directory: C:\DRIVER.1.38.log  
Machine: CL19  
Parameter Set: Audit  
Index: 925000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.38  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp37

Name: DRIVER.1.39  
Description:  
Directory: C:\DRIVER.1.39.log  
Machine: CL20  
Parameter Set: Audit  
Index: 950000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.39  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp38

Name: DRIVER.1.40  
Description:  
Directory: C:\DRIVER.1.40.log  
Machine: CL20  
Parameter Set: Audit  
Index: 975000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.40  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10

CLIENT\_NURAND: 233  
Additional Options: spprefix=sp39

Name: DRIVER.1.41  
Description:  
Directory: C:\DRIVER.1.41.log  
Machine: CL21  
Parameter Set: Audit  
Index: 1000000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.41  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp40

Name: DRIVER.1.42  
Description:  
Directory: C:\DRIVER.1.42.log  
Machine: CL21  
Parameter Set: Audit  
Index: 1025000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.42  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp41

Name: DRIVER.1.43  
Description:  
Directory: C:\DRIVER.1.43.log  
Machine: CL22  
Parameter Set: Audit  
Index: 1050000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.43  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233

CPU: 0  
Additional Options: spprefix=sp42

Name: DRIVER.1.44  
Description:  
Directory: C:\DRIVER.1.44.log  
Machine: CL22  
Parameter Set: Audit  
Index: 1075000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.44  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp43

Name: DRIVER.1.45  
Description:  
Directory: C:\DRIVER.1.45.log  
Machine: CL23  
Parameter Set: Audit  
Index: 1100000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.45  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp44

Name: DRIVER.1.46  
Description:  
Directory: C:\DRIVER.1.46.log  
Machine: CL23  
Parameter Set: Audit  
Index: 1125000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.46  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1

Additional Options: spprefix=sp45  
Name: DRIVER.1.47  
Description:  
Directory: C:\DRIVER.1.47.log  
Machine: CL24  
Parameter Set: Audit  
Index: 1150000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.47  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp46

Name: DRIVER.1.48  
Description:  
Directory: C:\DRIVER.1.48.log  
Machine: CL24

Parameter Set: Audit  
Index: 1175000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.48  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp47

Name: DRIVER.1.49  
Description:  
Directory: C:\DRIVER.1.49.log  
Machine: CL25  
Parameter Set: Audit  
Index: 1200000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.49  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp48

Name: DRIVER.1.50  
Description:  
Directory: C:\DRIVER.1.50.log  
Machine: CL25  
Parameter Set: Audit  
Index: 1225000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.50  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp49

Name: DRIVER.1.51  
Description:  
Directory: C:\DRIVER.1.51.log  
Machine: CL26  
Parameter Set: Audit

Index: 1250000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.51  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp50

Name: DRIVER.1.52  
Description:  
Directory: C:\DRIVER.1.52.log  
Machine: CL26  
Parameter Set: Audit  
Index: 1275000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.52  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp51

Name: DRIVER.1.53  
Description:  
Directory: C:\DRIVER.1.53.log  
Machine: CL27  
Parameter Set: Audit  
Index: 1300000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.53  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp52

Name: DRIVER.1.54  
Description:  
Directory: C:\DRIVER.1.54.log  
Machine: CL27  
Parameter Set: Audit  
Index: 1325000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.54  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp53

Name: DRIVER.1.55  
Description:  
Directory: C:\DRIVER.1.55.log  
Machine: CL28  
Parameter Set: Audit  
Index: 1350000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.55  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp54

Name: DRIVER.1.56

Description: C:\DRIVER.1.56.log  
Machine: CL28  
Parameter Set: Audit  
Index: 1375000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.56  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp55

Name: DRIVER.1.57  
Description:  
Directory: C:\DRIVER.1.57.log  
Machine: CL29  
Parameter Set: Audit  
Index: 1400000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.57  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp56

Name: DRIVER.1.58  
Description:  
Directory: C:\DRIVER.1.58.log  
Machine: CL29  
Parameter Set: Audit  
Index: 1425000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.58  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp57

Name: DRIVER.1.59  
Description:

Machine: CL30  
Directory: C:\DRIVER.1.59.log  
Parameter Set: Audit  
Index: 1450000000  
Seed: 73660  
Configured Users: 14630  
Pipe Name: DRIVER.1.59  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp58

Name: DRIVER.1.60  
Description:  
Directory: C:\DRIVER.1.60.log  
Machine: CL30  
Parameter Set: Audit  
Index: 1475000000  
Seed: 73660  
Configured Users: 14630

Pipe Name: DRIVER.1.60  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp59

Name: DRIVER.1.61  
Description:  
Directory: C:\DRIVER.1.61.log  
Machine: CL31  
Parameter Set: Audit  
Index: 1500000000  
Seed: 73660  
Configured Users: 10970  
Pipe Name: DRIVER.1.61  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp60

Name: DRIVER.1.62  
Description:  
Directory: C:\DRIVER.1.62.log

Machine: CL31  
Parameter Set: Audit  
Index: 1525000000  
Seed: 73660  
Configured Users: 10970  
Pipe Name: DRIVER.1.62  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp61

Name: DRIVER.1.63  
Description:  
Directory: C:\DRIVER.1.63.log  
Machine: CL32  
Parameter Set: Audit  
Index: 1550000000  
Seed: 73660  
Configured Users: 10970  
Pipe Name: DRIVER.1.63  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 0  
Additional Options: spprefix=sp62

Name: DRIVER.1.64  
Description:  
Directory: C:\DRIVER.1.64.log  
Machine: CL32  
Parameter Set: Audit  
Index: 1575000000  
Seed: 73660  
Configured Users: 10970  
Pipe Name: DRIVER.1.64  
Connect Rate: 0  
Start Rate: 0  
Max. Concurrency: -1  
Concurrency Rate: 10  
CLIENT\_NURAND: 233  
CPU: 1  
Additional Options: spprefix=sp63

Number of User groups: 64

Driver Engine: DRIVER.1.01  
IIS Server: DL1

Database: tpcc  
SQL Server: via:sqlorca  
User: sa  
Protocol: HTML  
w\_id Range: 1 - 1433  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14330  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.02  
IIS Server: DL1  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 1434 - 2867  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14340

District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.03  
IIS Server: DL2  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 2868 - 4301  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14340  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.04  
IIS Server: DL2  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 4302 - 5735  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14340  
District id: 1

Scale Down: No  
Driver Engine: DRIVER.1.05  
IIS Server: DL3  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 5736 - 7183  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14480  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.06  
IIS Server: DL3  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 7184 - 8631

w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14480  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.07  
IIS Server: DL4  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 8632 - 10079  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14480  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.08  
IIS Server: DL4  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 10080 - 11527  
w\_id Min Warehouse: 1

w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14480  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.09  
IIS Server: DL5  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 11528 - 12990  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.10  
IIS Server: DL5  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 12991 - 14453  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.11  
IIS Server: DL6  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 14454 - 15916  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.12  
IIS Server: DL6  
SQL Server: via:sqlorca  
Database: tpcc

Protocol: HTML  
w\_id Range: 15917 - 17379  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.13  
IIS Server: DL7  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 17380 - 18842  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.14  
IIS Server: DL7  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 18843 - 20305  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.15  
IIS Server: DL8  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 20306 - 21768  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.16  
IIS Server: DL8  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 21769 - 23231  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.17  
IIS Server: DL9  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 23232 - 24694  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991

Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.18  
IIS Server: DL9  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 24695 - 26157  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.19  
IIS Server: DL10  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 26158 - 27620  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal

User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.20  
 IIS Server: DL10  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 27621 - 29083  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.21  
 IIS Server: DL11  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 29084 - 30546  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.22  
 IIS Server: DL11  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 30547 - 32009  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.23  
 IIS Server: DL12  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML

w\_id Range: 33472  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.24  
 IIS Server: DL12  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 33473 - 34935  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.25  
 IIS Server: DL13  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 34936 - 36398  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.26  
 IIS Server: DL13  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 36399 - 37861  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.27  
 IIS Server: DL14

Database: tpcc  
 SQL Server: via:sqlorca  
 User: sa  
 Protocol: HTML  
 w\_id Range: 37862 - 39324  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.28  
 IIS Server: DL14  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 39325 - 40787  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.29  
 IIS Server: DL15  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 40788 - 42250  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1  
 Scale Down: No

Driver Engine: DRIVER.1.30  
 IIS Server: DL15  
 SQL Server: via:sqlorca  
 Database: tpcc  
 User: sa  
 Protocol: HTML  
 w\_id Range: 42251 - 43713  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 14630  
 District id: 1



Scale Down: No

Driver Engine: DRIVER.1.31  
IIS Server: DL16  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 43714 - 45176  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.32  
IIS Server: DL16  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 45177 - 46639  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.33  
IIS Server: DL17  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 46640 - 48102  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.34  
IIS Server: DL17  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 48103 - 49565  
w\_id Min Warehouse: 1

Scale: Normal  
w\_id Max Warehouse: 91991  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.35  
IIS Server: DL18  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 49566 - 51028  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.36  
IIS Server: DL18  
SQL Server: via:sqlorca

Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 51029 - 52491  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.37  
IIS Server: DL19  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 52492 - 53954  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.38  
IIS Server: DL19  
SQL Server: via:sqlorca  
Database: tpcc

Protocol: HTML  
w\_id Range: 53955 - 55417  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.39  
IIS Server: DL20  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 55418 - 56880  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.40  
IIS Server: DL20  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 56881 - 58343  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.41  
IIS Server: DL21  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 58344 - 59806  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.42  
IIS Server: DL21  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 59807 - 61269  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.43  
IIS Server: DL22  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 61270 - 62732  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.44  
IIS Server: DL22  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 62733 - 64195  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.45  
IIS Server: DL23  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 64196 - 65658  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal

District id: 14630  
Scale Down: No

Driver Engine: DRIVER.1.46  
IIS Server: DL23  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 65659 - 67121  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.47  
IIS Server: DL24  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa

Protocol: HTML  
w\_id Range: 67122 - 68584  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.48  
IIS Server: DL24  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 68585 - 70047  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.49  
IIS Server: DL25  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML

w\_id Min Warehouse: 71510  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.50  
IIS Server: DL25  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 71511 - 72973  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.51

IIS Server: DL26  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 72974 - 74436  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.52  
IIS Server: DL26  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 74437 - 75899  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.53  
IIS Server: DL27

SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 75900 - 77362  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.54  
IIS Server: DL27  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 77363 - 78825  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.55  
IIS Server: DL28  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 78826 - 80288  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.56  
IIS Server: DL28  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 80289 - 81751  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1

Scale Down: No  
Driver Engine: DRIVER.1.57  
IIS Server: DL29  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 81752 - 83214  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.58  
IIS Server: DL29  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 83215 - 84677

w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.59  
IIS Server: DL30  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 84678 - 86140  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.60  
IIS Server: DL30  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 86141 - 87603  
w\_id Min Warehouse: 1

Scale Normal  
w\_id Max Warehouse: 91991  
User Count: 14630  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.61  
IIS Server: DL31  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 87604 - 88700  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 10970  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.62  
IIS Server: DL31  
SQL Server: via:sqlorca

Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 88701 - 89797  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 10970  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.63  
IIS Server: DL32  
SQL Server: via:sqlorca  
Database: tpcc  
User: sa  
Protocol: HTML  
w\_id Range: 89798 - 90894  
w\_id Min Warehouse: 1  
w\_id Max Warehouse: 91991  
Scale: Normal  
User Count: 10970  
District id: 1  
Scale Down: No

Driver Engine: DRIVER.1.64  
IIS Server: DL32  
SQL Server: via:sqlorca  
Database: tpcc

User: sa  
 Protocol: HTML  
 w\_id Range: 90895 - 91991  
 w\_id Min Warehouse: 1  
 w\_id Max Warehouse: 91991  
 Scale: Normal  
 User Count: 10970  
 District id: 1  
 Scale Down: No

Number of Parameter Sets: 3

Delay	Weight	Time	Time	Delay	Fence
	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			

AuditShutdown

Menu	Txn	Think	Key	RT	RT
Delay	Weight	Time	Time	Delay	Fence
	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

Menu	Txn	Think	Key	RT	RT
Delay	Weight	Time	Time	Delay	Fence
	New Order	44.92	12.05		18.01
0.10	5.00	0.10			
	Payment	43.02	12.05		3.01
0.10	5.00	0.10			
	Delivery	4.02	5.05		2.01
0.10	5.00	0.10			
	Stock Level	4.02	5.05		2.01
0.10	20.00	0.10			
	Order Status	4.02	10.05		2.01
0.10	5.00	0.10			

~Default

Menu	Txn	Think	Key	RT	RT
------	-----	-------	-----	----	----

## Appendix D 60 Day Space Requirements

TPC-C 60 Day Space Requirements						
Warehouses	98000				TpmC	1,082,202.00
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	98000	10688	12008	1,135		23831
District	980000	110512	12152	6,133		128797
Customer	2940000000	2138181888	133568192	113,587,504		2385337584
History	2951377532	177643856	1379464		30,430,173	179023320
NewOrder	881863341	21748352	86872	1,091,761		22926985
Orders	2951275705	103837856	458808		1,062,849	104296664
OrderLine	29512686460	1980669064	9271688		603,951,739	1989940752
Item	100000	9528	512	502		10542
Stock	9800000000	3136000000	6779968	157,138,998		3299918966
Total		7,558,211,744	151,569,664	271,826,034	635,444,761	7,981,607,442
	MB					
Dynamic Space	2,209,132	Sum of Data for Order, Orderline and History				
Static Space	5,585,407	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - ( Dynamic + Static Space)				
Daily Growth	390,323	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	29,004,770					
60 Day Space GB	28,324.97	GB				
Log Size	500,000.00	MB				
KB Per New Order	6.50	KB				
8 hr log MB	3,298,048	MB				
8 hr log GB	3,220.7495	GB				
		Disks	Disks	Formatted Size	Space	
Space Usage	GB Needed	Measured	Size	Size	Available	
180 Day Space DB	28,324.97	616	18GB	16.900	10410.40	
		1064	36	33.500	35644.00	
			4GB	3.999	0.00	
Total DB					<b>46,054.40</b>	
8-hr log + mirror	6,441.4991	56	146GB	136.700	<b>7,655.20</b>	
OS, Swap	3	2	18GB	16.900	33.80	
		2	36GB	33.500	67.00	
Total Storage	34,769.47	GB			53,743.40	

## Appendix E 3<sup>rd</sup> Party Pricing

Microsoft Corporation      Tel 425 882 8080  
One Microsoft Way      Fax 425 936 7329  
Redmond, WA 98052-6399      <http://www.microsoft.com/>

*Microsoft*

May 23, 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
810-00846	<b>SQL Server 2005 Enterprise Edition</b> <i>Per Processor Licensing</i> <i>Discount Schedule: Open Program - Level C</i> <i>Unit Price reflects a 15% discount from the retail unit price of \$23,424.</i>	\$19,909	64	\$ 1,274,176
P73-00295	<b>Windows Server 2003, Standard Edition</b> <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 28% discount from the retail unit price of \$999.</i>	\$719	32	\$23,008
254-00170	<b>Visual C++ Standard Edition</b> <i>Discount Schedule: No Discounts Applied</i>	\$109	1	\$109
	<b>Microsoft Problem Resolution Services</b> <i>Professional</i> <i>(1 incident)</i>	\$245	1	\$245

Some products may not be currently orderable but will be available through Microsoft's normal distribution channels by November 8, 2005.

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](mailto:jamiere@microsoft.com).

Reference ID: PCguzi0523053435.

Please include this Reference ID in any correspondence regarding this price quote.



We will meet or beat any competitor's price. [see details](#)

Free UPS Ground Shipping Everyday (all orders over \$ 50.00, US Only)

[Order Tracking](#)  
[Shopping Cart](#)

Brands [KVM Switches](#) [KVM Stuff](#) [KVM Cables](#) [Video Stuff](#) [Remote Reboot](#) [Digital KVM](#)

## Shopping Cart

Delete	Accessories	No.	Product	Quantity	Price	Extended
	<a href="#">Add-On</a>	5150	<a href="#">Startech StarView 1 X 32 KVM Switch w/OSD</a>	1	\$ 1599.00	1,599.00
	<a href="#">Add-On</a>	5250	<a href="#">KVM Premium 6ft. PS2 3-in-1 Cable Kit</a>	32	\$ 12.00	384
Sub Total						<b>1,983.00</b>

### Contact

Toll-Free:  
**1-800-KVM-Switch**  
( 1-800-566-7948)  
Int'l/local:  
1.770.971.1924  
Fax:  
1.770.971.2639  
Email:  
[sales@kvms.com](mailto:sales@kvms.com)  
Mail:  
PO Box 669875  
Marietta, GA 30066-0115

Check-Out



### XpressTrack

Enter XpressTrack Code (if any)

GO

### PromoCode

Enter promotional Code (if any)

GO

### Instructions

- To increase the quantity, just select the quantity in the quantity box.
- 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-Free 1.888.566.7948 and reference this code **POR-322510**

### Payments Accepted

- **Credit Cards**  
Visa, Master Card, Amex and Discover.

<http://www.kvms.com/shop/shop.asp>

6/1/2005

**SAN Marketplace** **Where the Storage Professionals Shop!** **1-866-463-3372**  
*Call us Toll-Free*

[Home](#) | [About Us](#) | [Support](#) | [Search](#) | [Checkout](#) | [Contact Us](#)

Wednesday June 1st 2005

Shopping cart  
 3 Product(s) in cart  
 9 Prime-SB5200  
 3 SB5200-16A  
 1 Finisar FTRJ-8519 2Gb SFP  
 Total \$25,142.88  
[> Checkout](#)

Details	Name	Unit Price	Quantity	Total	Select
<b>Prime-SB5200</b>	Prime-SB5200	\$1,327.32	<input type="text" value="9"/>	\$11,945.88	<input type="checkbox"/>
<b>SB5200-16A</b>	SB5200-16A	\$4,371.00	<input type="text" value="3"/>	\$13,113.00	<input type="checkbox"/>
<b>FTRJ8519P1BNL</b>	Finisar FTRJ-8519 2Gb SFP	\$84.00	<input type="text" value="53"/>	\$4,452.00	<input type="checkbox"/>

**Grand Total:** \$29,510.88 [Delete](#)

[Continue Shopping](#) [Update Totals](#)

Press checkout to enter your customer and shipping information.




- Call Us Toll-Free*  
**1-866-463-3372**
- Switches**
- [QLogic](#)
  - [McData](#)
  - [EMC](#)
  - [Brocade](#)
- Host Bus Adapters**
- [AMCC/INI](#)
  - [Emulex](#)
  - [QLogic](#)
- Fibre Cables**
- [LC-LC Cables](#)
  - [LC-SC Cables](#)
  - [SC-SC Cables](#)
  - [HSSDC/2 Cables](#)
  - [DB9 Cables](#)
- SFPs & GBICs**
- [Finisar](#)
- Other**
- [Credit Application](#)



Copyright © 2004



 <b>Hewlett Packard Company</b>	<b>HP Integrity Superdome Enterprise Server</b>			<b>TPC-C Rev. 5.0</b>			
				Report Date: April, 2003			
Description	Price Key	Part Numbr	Unit Price	Qty	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 (LP9802)	1	AB232A	\$1,550	4	\$6,200		
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	1682	\$502,918		
Storageworks MSA30 SB (incl boot)	1	302969-B21	\$2,829	121	\$342,309		
Storage MSA30 Support 3yrs 24x7 4 hr incl disks	1	HA110A3 #7GT	\$1,827	121		\$221,067	
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 MAS1000 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		
<b>Server Subtotal</b>					<b>\$5,594,466</b>	<b>\$1,122,614</b>	
Microsoft Windows Server 2003, Datacenter Edition (64-bit)	1	T2372A opt064	\$128,000	1	\$128,000		
SP1							
Microsoft SQL Server 2005 Enterprise Edition 64-bit	<b>Server Software Subtotal</b>					<b>\$128,000</b>	<b>\$0</b>
DL140 Dual 3.06GHz Xeon 1GB 80GB ATA	1	356533-B21	\$1,799	32	\$57,568		
HP 3 Yr. 4h 24x7 ProLiant DL140 Hardware Support	1	U9510E	\$419	12		\$5,028	
1GB PC2100 ECC DDR	1	351109-B21	\$419	32	\$13,408		
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	37	\$46,472		
5M LC to LC Cable Kit	1	221692-B22	\$82	48	\$3,936		
<b>Client Subtotal</b>					<b>\$121,590</b>	<b>\$5,028</b>	
HP's Large Configuration Discount *	<b>* Discounts:</b>					<b>-\$2,082,351</b>	<b>-\$398,336</b>
<b>Total:</b>					<b>\$3,761,705</b>	<b>\$729,306</b>	
<b>* All discounts are based on US list prices and for similar quantities and configurations</b>					<b>3 year cost of ownership: \$4,491,011</b>		