



i n v e n t

TPC Benchmark® C Full Disclosure Report

HP ProLiant BL685c G7

using Microsoft SQL Server 2005 Enterprise X64 Edition SP3
on Microsoft Windows Server 2008 R2 Enterprise Edition for
X64-Based Systems

First Edition
May 23, 2011

First Edition - May 23, 2011

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 2011.

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., May 23, 2011

HP, HP ProLiant 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 ProLiant BL685c G7 in a client/server configuration, using Microsoft SQL Server 2005 Enterprise X64 Edition SP3 and Microsoft COM+ Transaction Monitor. The operating system used for the benchmark was Microsoft Windows Server 2008 R2 Enterprise Edition for X64-Based Systems .

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

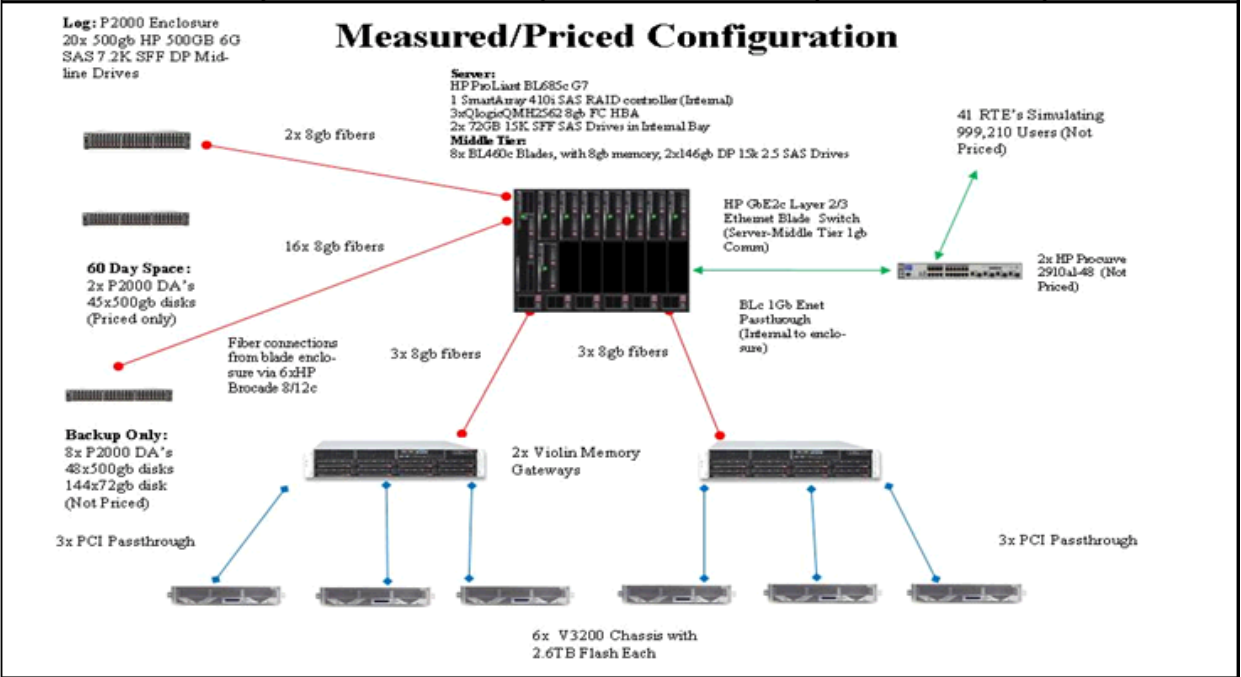
The following pages contain the executive summary of the benchmark results for the HP ProLiant BL685c G7 system. The Standard System Summary is given below.

Company Name	System Name	Database Software	Operating System
Hewlett-Packard Company	HP ProLiant BL685c G7	Microsoft SQL Server 2005 Enterprise X64 Edition SP3	Microsoft Windows Server 2008 R2 Enterprise Edition for X64-Based Systems
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$642,516 USD	1,263,599 tpmC	\$0.51 USD per tpmC	May 23, 2011

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 ProLiant BL685c G7			TPC-C Rev 5.11
				Pricing Rev 1.6
Total System Cost	TPC Throughput	Price/Performance	Report Date	Availability Date
\$642,516 USD	1,263,599 tpmC	\$0.51 USD per tpmC	May 23, 2011	May 23, 2011
Procs/Cores/Threads	DataBase Manager	Operating System	Other Software	Number of Users
Srvr - 4/48/48 AMD Opteron 6174 @ 2.2GHz Client - 8 x 2/8/8 Intel Xeon E5520 @ 2.26 GHz	Microsoft SQL Server 2005 Enterprise X64 Edition SP3	Microsoft Windows Server 2008 R2 Enterprise Edition for X64-Based Systems	Microsoft Visual C++ Microsoft COM+ Transaction Monitor	999,210



System Components	Server		Each Client	
	Qty	Type	Qty	Type
Procs/Cores/Thrds	4/48/48	AMD Opteron 6174 CPUs @ 2.2GHz	2/8/8	2.26 GHz Intel Xeon E5520
Cache Memory		6 MB L3 cache		8 Mbyte L3 Cache
Memory	32	16 Gbyte	1	8196 MB
Disk Controllers	3	QLogic QMH2562 8Gb DP FC Mezzanine Ca	1	SA 410i
	1	Smart Array P410i		
Disk Drives	6	Violin Memory Appliance	2	146 Gbyte disk
	20	HP 500GB 6G SAS 7.2K SFF Dual Port Hard		
	45	HP 500GB 6G SAS 7.2K SFF Dual Port Hard Dr (60 Day)		
	2	HP 72GB 15K SAS drives (OS)		
Total Storage		39709.11		146 Gbyte
Terminals	1	Console Terminal	1	Console Terminal



HP ProLiant BL685c G7

TPC-C	5.11
TPC-Pricing	1.6.0
Report date	May 23, 2011
Availability Date	May 23, 2011

Description	Part Number	Brand	Unit	Qty.	Extended	3 Yr Maint
Server Hardware (Tier B)						
HP BL685c G7 CTO Blade (support included under client hardware)	518878-B21	1	6,251	1	6,251	
HP BL685c G7 O6174 12C 2P Kit	518871-B21	1	2,679	1	2,679	
HP BL685c G7 O6174 12C 2P FIO Kit	518871-L21	1	2,679	1	2,679	
HP 16GB (1x16GB) Quad Rank x4 PC3-8500 (DDR3-1066) Registered CAS-7 Memory Kit	593915-B21	1	899	32	28,768	
HP 72GB SAS 15K SFF DP ENT HDD	512545-B21	1	279	2	558	
QLogic QMH2562 8Gb FC HBA for HP BladeSystem c-Class	451871-B21	1	849	3	2,547	
HP LE1851w 18.5-Inch wide Monitor	NK033AA#ABA	1	159	1	159	
HP PS/2 Keyboard And Mouse Bundle	RC464AA#ABA	1	39	1	39	
			Subtotal		\$43,680	\$0
Server Software						
Microsoft SQL Server 2005 Enterprise X64 Edition	810-03134	2	23,432	4	93,728	Incl Below
Windows Server 2008 R2 Enterprise Edition	P72-04217	2	2,280	1	2,280	Incl Below
Microsoft Problem Resolution Services	N/A	2	259	1		259
			Subtotal		\$96,008	259
Storage						
V-3200 Chassis with PCIe 20 Gbit/s Interface 2.6TB Flash (21 x VIMM-128GSLC)	V-3202	3	50,250	4	201,000	
V-3200 Chassis with PCIe 20 Gbit/s Interface 2.6TB Flash (42 x VIMM-64GSLC)	V-3202A	3	50,250	2	100,500	
Violin Memory Gateway: 4 Port 4/8 Gbit/s Fiber Channel Interfaces	VMG-8GFCx4	3	8,040	2	16,080	
Violin Premium Maintenance - 1 year 4h 24 x7 support	VS-PM1Y	3	18,960	3		56,880
HP StorageWorks P2000 G3 FC MSA Dual Controller SFF Modular Smart Array	AP846A	1	10,350	1	10,350	
HP 3 years 4 hours 24x7 MSA 2000 G3 HW Support	UV394E	1	3,265	1		3,265
HP 500GB 6G SAS 7.2K SFF (2.5-inch) Dual Port Midline Hard Dr (Log)	507610-B21	1	349	20	6,980	
HP StorageWorks P2000 G3 FC MSA Dual Controller SFF Modular Smart Array	AP846A	1	10,350	2	20,700	
HP 3 years 4 hours 24x7 MSA 2000 G3 HW Support	UV394E	1	3,265	2		6,530
HP 500GB 6G SAS 7.2K SFF (2.5-inch) Dual Port Midline Hard Dr	507610-B21	1	349	45	15,705	
			Subtotal		371,315	66,675
Client Hardware (Tier A)						
HP BLc7000 Blade Enclosure 3 Inch LCD display	507019-B21	1	4,837	1	4,837	
Single Phase Power option for Blade Enclosure	413379-B21	1	175	1	175	
HP BLc 6x Active Cool 200 FIO Fan Option	517520-B21	1	894	1	894	
HP 2400W High Efficiency Power Supply	499243-B21	1	349	6	2,094	
HP BLc 1Gb Enet Pass Thru Mod Opt Kit	406740-B21	1	999	1	999	
HP GbE2c Layer 2/3 Ethernet Blade Switch	438030-B21	1	1,799	1	1,799	
HP ProLiant BL460C G6 E5520 2.26Ghz Quad Core Blade Server	507782-B21	1	2,489	8	19,912	
HP E5520 BL460c G6 FIO Kit	507799-B21	1	599	8	4,792	
HP 4GB 2Rx4 PC3-10600R-9 Memory Kit	500658-B21	1	189	16	3,024	
HP 146GB 15k 2.5 dual Port HP SAS Drive	512547-B21	1	369	16	5,904	
HP 1year 4h 24x7 B Series Blades HW Support 16 Blades	UB329E	1	612	3		1,836
			Subtotal		44,430	1,836
Client Software						
Microsoft Windows Server 2008 Standard (x64) R2	P73-04980	2	711	8	5,688	
			Subtotal		5,688	0
Infrastructure						
HP 10642 G2 Pallet Universal Rack	AF001A	1	1,249	2	2,498	
HP Brocade 8/12c SAN Switch for BladeSystem c-Class	AJ820A	1	6,374	6	38,244	
HP 8Gb Shortwave B-series Fibre Channel 1 Pack SFP+ Transceiver	AJ716A	1	199	12	2,388	
2 m LC-LC Multi-Mode Fibre Channel Cable	221692-B21	1	75	12	900	
2 m LC-LC Multi-Mode Fibre Channel Cable (spares)	221692-B21	1	75	3	225	
2.1m/7ft CAT5 RJ45 M/M Ethernet Cable	C7535A	1	2	2	3	
2.1m/7ft CAT5 RJ45 M/M Ethernet Cable (spares)	C7535A	1	2	2	3	
			Subtotal		44,261	0
Total Extended Price					\$605,382	\$68,770
Total Discounts					\$29,776	\$1,861
Grand Total					\$575,606	\$66,909
Pricing: 1=HP Direct 800-203-6748 2= Microsoft. 3=Violin Memory Systems Note 1: Discount based on HP Direct guidance applies to all lines where pricing = 1. Note 2: All the hardware are available to order. Note 3: The benchmark results were audited by Lorna Livingtree of Performance Metrics.					Three-year Cost of Ownership: USD \$642,516 TPM-C 1,263,599 \$ USD/TPM-C \$0.51	

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 at pricing@tpc.org. Thank you.

Numerical Quantities Summary for HP ProLiant BL685c G7

MQTH, Computed Maximum Qualified Throughput

1,263,599 tpmC

Response Times (in seconds)

	90th %-ile	Maximum	Average
New-Order	0.26s	5.32s	0.15s
Payment	0.25s	5.39s	0.14s
Order-Status	0.26s	5.01s	0.15s
Delivery (interactive portion)	0.18s	2.12s	0.13s
Delivery (deferred portion)	0.07s	2.43s	0.03s
Stock-Level	0.33s	2.16s	0.20s
Menu	0.18s	2.91s	0.13s

Response time delay added for emulated components 0.1 seconds

Transaction Mix, in percent of total transactions

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

Keying/Think Times

	Keying Time			Think Time		
	Min	Avg	Max	Min	Avg	Max
New-Order	18.00s	18.02s	18.41s	0.00s	12.06s	120.53s
Payment	3.00s	3.02s	3.41s	0.00s	12.06s	120.53s
Order-Status	2.00s	2.02s	2.4s	0.00s	10.05s	100.52s
Delivery (interactive)	2.00s	2.02s	2.4s	0.00s	5.07s	50.52s
Stock-Level	2.00s	2.02s	2.41s	0.00s	5.07s	50.52s

Test Duration

Ramp up time	57 minutes
Measurement interval	120 minutes
Transactions during measurement interval	350711596
Ramp down time	46 minutes

Checkpointing

Number of checkpoints in measurement interval	4
Checkpoint Interval	29.17 minutes

Table of Contents

Abstract.....	3
Overview.....	3
TPC Benchmark® C Metrics.....	3
Standard and Executive Summary Statements.....	3
Auditor.....	3
Preface.....	9
Document Structure.....	9
TPC Benchmark® C Overview.....	9
System Overview.....	10
General Items.....	11
Test Sponsor.....	11
Application Code and Definition Statements.....	11
Parameter Settings.....	11
Configuration Diagrams.....	11
Chapter 1 Logical Database Design.....	13
1.1 Table Definitions.....	13
1.2 Physical Organization of the Database.....	13
1.3 Insert and Delete Operations.....	13
1.4 Partitioning.....	13
1.5 Replication, Duplication or Additions.....	13
Chapter 2 Transaction and Terminal Profiles.....	14
2.1 Random Number Generation.....	14
2.2 Input/Output Screen Layout.....	14
2.3 Priced Terminal Feature Verification.....	14
2.4 Transaction Statistics.....	14
2.5 Presentation Manager or Intelligent Terminal.....	15
2.6 Queuing Mechanism.....	15
Chapter 3 Transaction and System Properties.....	16
3.1 Transaction System Properties (ACID Tests).....	16
3.2 Atomicity Tests.....	16
3.2.1 COMMIT Transaction.....	16
3.2.2 ROLLBACK Transaction.....	16
3.3 Consistency Tests.....	16
3.4 Isolation Tests.....	17
3.5 Durability Tests.....	17
3.5.1 Loss of Data.....	17
3.5.2 Loss of System / Memory and loss of Log.....	18
Chapter 4 Scaling and Database Population.....	19
4.1 Database Layout.....	19
4.2 Initial Cardinality of Tables.....	22
4.3 60 Day Space.....	22
4.3.1 Transaction Log Space Requirements.....	22
4.4 Type of Database Used.....	23
4.5 Database Mapping.....	23
Chapter 5 Performance Metrics and Response Time.....	24
5.1 Throughput.....	24
5.2 Response Times.....	24
5.3 Keying and Think Times.....	24
5.4 Response Time Frequency.....	25
5.4.1 New Order Response Time.....	25
5.4.2 Payment Response Time Distribution.....	26

5.4.3	Order Status Response Time.....	27
5.4.4	Delivery Response Time Distribution.....	28
5.4.5	Stock Level Response Time.....	29
5.4.6	Response Time Versus Throughput.....	30
5.4.7	New Order Think Time Distribution.....	31
5.4.8	Throughput Versus Time Distribution.....	32
5.5	Steady State Determination.....	32
5.6	Work Performed During Steady State.....	32
5.6.1	Checkpoint.....	33
5.6.2	Checkpoint Conditions.....	33
5.6.3	Checkpoint Implementation.....	33
5.7	Measurement Period Duration.....	33
5.8	Regulation of Transaction Mix.....	33
5.9	Transaction Mix.....	33
5.10	Transaction Statistics.....	34
5.11	Checkpoint Count and Location.....	34
Chapter 6	SUT, Driver and Communications Definition.....	35
6.1	RTE Description.....	35
6.2	Emulated Components.....	35
6.3	Functional Diagram.....	35
6.4	Networks.....	35
6.5	Operator Intervention.....	35
Chapter 7	Pricing.....	36
7.1	System Pricing.....	36
7.2	General Availability, Throughput and Price Performance.....	36
7.3	Country Specific Pricing.....	36
7.4	Usage Pricing.....	36
7.5	Testing.....	37
Chapter 8	Audit.....	38
8.1	Auditor's Information.....	38
Appendix A	Source Code.....	41
Appendix B	Database Load.....	172
B.1	Database Options.....	194
B.2	Table definitions.....	195
B.3	Stored Procedures.....	197
Appendix C	Tunable Parameters.....	231
C.1	Microsoft SQL Server 8.0 Configuration Parameters.....	267
C.2	Client System Configuration Parameters.....	268
	RTE Input Parameters.....	298
Appendix D	60 Day Space Requirements.....	315
Appendix E	3 rd Party Pricing.....	316
Appendix F	System Availability.....	318

Preface

Document Structure

This is the full disclosure report for a benchmark test of the HP ProLiant BL685c G7 using Microsoft SQL Server 2005 Enterprise X64 Edition SP3. It meets the requirements of the TPC Benchmark® C Standard Specification, Revision 5.11 dated February 2010. 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 ProLiant BL685c G7. 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 2008 R2 Enterprise Edition for X64-Based Systems and the database was Microsoft SQL Server 2005 Enterprise X64 Edition SP3, Version 9.00.2047.00.

The processor architecture of the HP ProLiant BL685c G7 was designed for the AMD Opteron 6174 processor. The HP ProLiant BL685c G7 used in this test was powered by 4 2.2GHz AMD Opteron 6174 processors, each with 6 MB of second level cache. The 4 processors each contain 12 cores. This configuration therefore presented 48 logical processors to the operating system.

This configuration used 512 GB of HP SDRAM.

The operating system, all executables and libraries, the master database, and swap space were contained in two 146 GB hard disks, attached to the internal HP SmartArray P410i and configured as a RAID1 boot pair. A partition was created on the same disc array as the log and was used for utility storage of scripts, the build environment, etc.

The database log storage was located on 1 HP StorageWorks MSA P2000 disk array. The MSA P2000 held 20 HP 500GB DP 2.5" 7.2K RPM SAS hard drives. The disk array was connected to the HP ProLiant BL685c G7 using a QLogic QMH2562 8Gb Fibre Channel Host Bus Adapter. The disks were configured as two RAID 1+0 arrays, and presented to the server as two LUN's, which were then software striped using the disk striping capabilities of the OS. The caches in the dual controllers in the P2000 were configured as an active-active mirroring pair.

The TPC-C database storage consisted of 6 Violin Memory Appliances. Each Violin Memory Appliance contained 2.6TB of flash memory. Two groups of three arrays were each connected to a memory gateway via PCI express over cable, which in turn output over Fibre Channel to the system under test. Windows partitions were created on each of the volumes to contain the CS and MISC SQL filegroups. The partition sizes were the same on all 6 volumes.

Each of the 8 clients was an HP BladeSystem C7000/DL460c with 2 Intel Xeon E5520 Processors at 2.26 GHz, 8196 MB RAM and two HP 146GB 15k 2.5 dual Port HP SAS Drives connected to the internal SmartArray P410i (mirrored), running Microsoft Windows Server 2008 Standard with IIS 7. Threading was disabled, so 8 logical processors were presented to the operating system.

The server and web-clients were networked together using standard Gigabit LAN connections from the web-clients to the blade switch, and from the switch to the server internal to the C7000 blade chassis. 41 remote terminal emulators (RTEs) emulated 999,210 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 used as the RTE emulators.

Microsoft SQL Server 2005 Enterprise X64 Edition SP3 was configured to utilize the default 8 hard NUMA nodes. No SoftNuma was used. Processor affinity was used to remove processor 0 from SQL server, and was used for network interrupts only. A script doing continuous checkpoints of 1750 seconds (29 minutes, 10 seconds) was started on the server.

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.

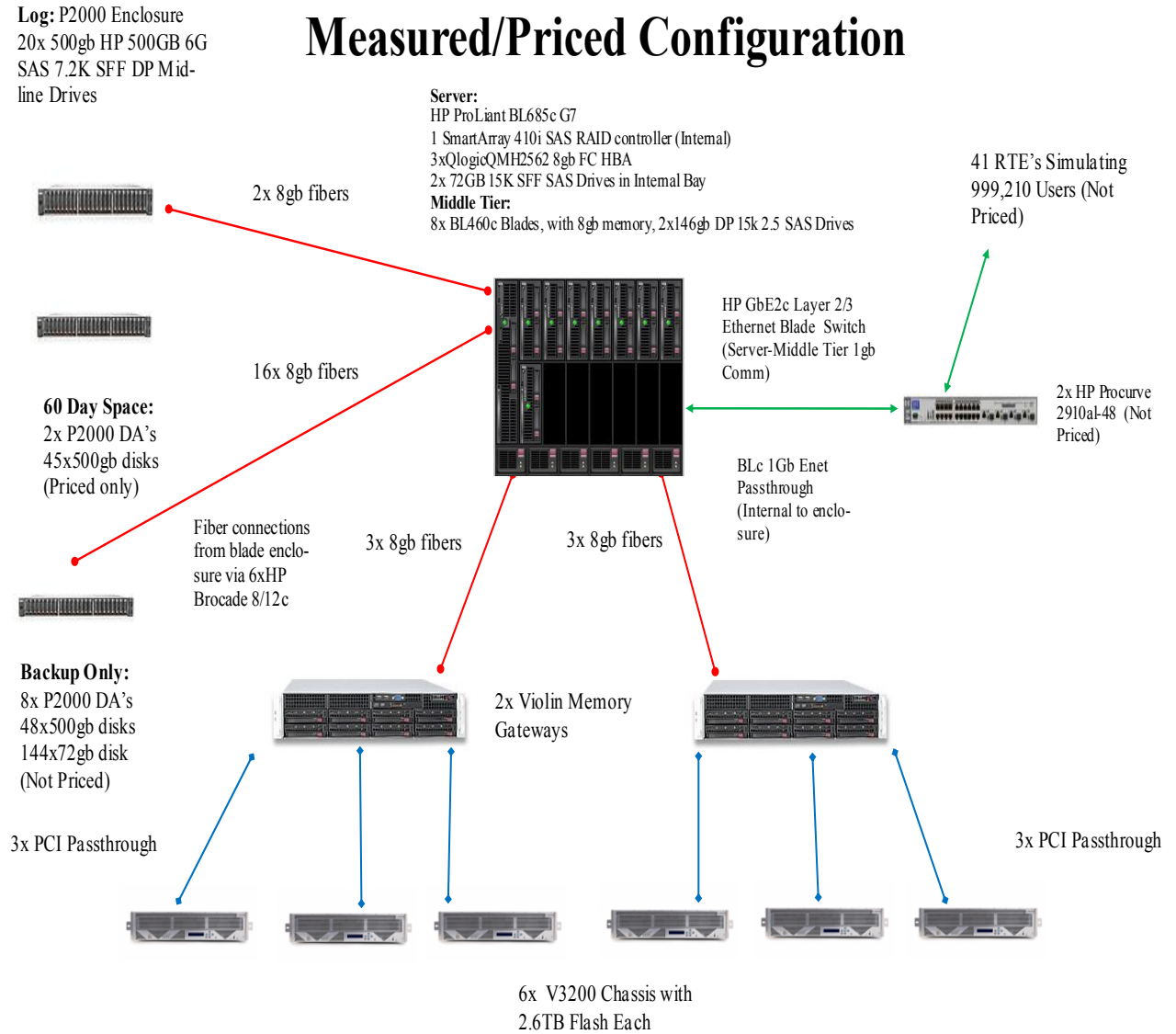
Appendix D contains the 60 day space calculations.

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



Priced configuration differed from Measured configuration in that 2 P2000s and 45 500GB disks were priced for 60 day space requirements only. In addition, the 8 P2000s connected for database backup were not active during the performance run.

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 priced database configuration used a total of 235 individual data devices, 6 Violin Memory System arrays, containing a total of 168 flash memory modules, 20 HP 500GB DP 2.5" 7.2K RPM SAS drives for log, and 45 HP 500GB DP 2.5" 7.2K RPM SAS(s) for 60 Day space, and 2 72GB drives for the OS.

The TPC-C database storage consisted of 6 Violin Memory Appliances. Each Violin Memory Appliance contained 2.6TB of flash memory. Windows partitions were created on each of the volumes to contain the CS and MISC SQL filegroups. Two CS and two MISC partitions were created on each of the the 6 volumes. The partition sizes were the same on all 6 volumes.

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	85.01%
	Remote warehouse	14.99%
	Non primary key access	60.01%
Order Status	Non primary key access	60.05%
Delivery	Skipped transactions	0
Transaction Mix	New Order	44.97%
	Payment	43.01%
	Delivery	4.01%
	Stock Level	4.01%
	Order Status	4.01%

2.5 Presentation Manager or Intelligent Terminal

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

Comment 1: *The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations, message bundling, and local storage of TPC-C rows.*

Comment 2: *This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

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

2.6 Queuing Mechanism

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

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

The source code is listed in Appendix A.

Chapter 3 Transaction and System Properties

3.1 Transaction System Properties (ACID Tests)

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

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

All of the specified ACID tests were successfully performed on the HP ProLiant BL685c G7, the System Under Test in this benchmark. 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 tests were performed on the full scale database..

3.5.1 Loss of Data

The standard driving mechanism was used to generate the transaction load of slightly more than 99,921 users for the test (10% of full load). To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The database was backed up using SQLServer backup facilities.
2. A sum of D_NEXT_O_ID was taken.
3. Slightly more than 99,921 (10%) users were logged in to the database and ran transactions.
4. After 5 minutes, one Violin memory module was removed from an array.
5. Since the Violin Memory Systems array is fully redundant and in a RAID configuration, there was no loss of data nor errors generated.
6. The RTE monitor was used to verify that no users were lost.
7. The workload was allowed to run for another five minutes.
8. The RTE was shutdown and a success file was created.

3.5.2 Loss of System / Memory and loss of Log

This was demonstrated on the full database with 100,000 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 999,210 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. 999,210 users were logged in to the database and ran transactions.
4. Rampup was performed until the TPMC rate was 90% of the reported rate.
5. After 5 minutes, one of the (mirrored) log disk was removed from the system, processing transactions continued.
6. After another 5 minutes, the system was powered off using the build in Integrated Lights Out (ILO) capability.. This reset the hardware, reran memory initialization, and reloaded the Windows OS.
7. The RTE continued running and completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
8. The RTE was stopped.
9. After Windows was finished booting, 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 6 Violin Memory Systems V3202 flash memory arrays that were connected to the system using QLogic QMH2562 8Gb Fibre Channel Host Bus Adapters. The HP Storage Works MSA P2000 disk array that held the database log was also connected to QLogic QMH2562 8Gb Fibre Channel Host Bus Adapter.

The priced database configuration used a total of 235 individual data devices, 6 Violin Memory System arrays, containing a total of 168 flash memory modules, 20 HP 500GB DP 2.5" 7.2K RPM SAS drives for log, and 45 HP 500GB DP 2.5" 7.2K RPM SAS(s) for 60 Day space, and 2 72GB drives for the OS.

Table 2a shows the complete data distribution.

Table 2a: Data Distribution

Controller Type	Disk #	Drives Enclosure RAID Lvl	Path Filesystem Partition	Size	Use
P410i Internal SmartArray	1	2x72GB SAS, Internal RAID1	C:, NTFS	72GB	Win2008 Boot, PageFile, Utility, Scripts Mount Point Root, DB Root File
Violin Memory Systems	1	21 X 128GB Violin intelligent memory modules	c:\mnt\cs\1\ (RAW) c:\mnt\misc\1\ (RAW) c:\mnt\cs\2\ (RAW) c:\mnt\misc\2\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files
Violin Memory Systems	1	21 X 128GB Violin intelligent memory modules	c:\mnt\cs\3\ (RAW) c:\mnt\misc\3\ (RAW) c:\mnt\cs\4\ (RAW) c:\mnt\misc\4\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files
Violin Memory Systems	1	21 X 128GB Violin intelligent memory modules	c:\mnt\cs\5\ (RAW) c:\mnt\misc\5\ (RAW) c:\mnt\cs\6\ (RAW) c:\mnt\misc\6\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files
Violin Memory Systems	1	21 X 128GB Violin intelligent memory modules	c:\mnt\cs\7\ (RAW) c:\mnt\misc\7\ (RAW) c:\mnt\cs\8\ (RAW) c:\mnt\misc\8\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files
Violin Memory Systems	1	42 X 64GB Violin intelligent memory modules	c:\mnt\cs\9\ (RAW) c:\mnt\misc\9\ (RAW) c:\mnt\cs\10\ (RAW) c:\mnt\misc\10\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files
Violin Memory Systems	1	42 X 64GB Violin intelligent memory modules	c:\mnt\cs\11\ (RAW) c:\mnt\misc\11\ (RAW) c:\mnt\cs\12\ (RAW) c:\mnt\misc\12\ (RAW) t:\ (NTFS) (striped)	500 GB 275 GB 500 GB 275 GB 49.9 GB	Customer/Stock FG MISC FG Customer/Stock FG MISC FG TempDB files

Table 2b: Log Distribution

QLogic QMH2562 8Gb Fibre Channel Host Bus Adapter			WINDOWS SERVER DISK MANAGER	
1 MSA P2000 Disk Array			Logical Disk 1 (Dynamic) Total= 4356 GB	
Slot ID	Controller		Partitions Dynamically Striped	
	vDisk 0 (RAID 1+0)	vDisk 1 (RAID 1+0)	Physical Disk 1	Physical Disk 2
0	500GB		L: RAW 2326.3GB (DB log)	L: RAW 2326.3GB (DB log)
1	500GB			
2	500GB			
3	500GB			
4	500GB			
5	500GB			
6	500GB			
7	500GB			
8	500GB			
9	500GB			
10	500GB			
11		500GB		
12		500GB		
13		500GB		
14		500GB		
15		500GB		
16		500GB		
17		500GB		
18		500GB		
19		500GB		
20		500GB		

4.2 Initial Cardinality of Tables

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

Table 3 shows the cardinality of the various tables.

Table 3: Table Cardinality

Table	Occurrences
Warehouse	100,000
District	1,000,000
Customer	3,000,000,000
History	3,000,000,000
Orders	3,000,000,000
New Orders	900,000,000
Order Line	3,000,000,000
Stock	10,000,000,000
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 3417.2 GB to sustain the log for 8 hours. Space available for the transaction log was 4653 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 X64 Edition SP3 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. cs_fg consists of 12 partitions at 500 GB each and misc_fg consist of 12 partitions at 275 GB each as shown in the createdb.sql. The log was configured with 400,000 MB at database creation, and was expanded to 1800 GB after database creation and load.

Chapter 5 Performance Metrics and Response Time

5.1 Throughput

Measured tpmC® must be reported.

Measured TpmC®: 1,263,599
Price per TpmC®: \$0.51 USD

5.2 Response Times

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

Table 3 shows the response times for all transaction types.

Table 4: Transaction Response Times

Response Times	Average	90th %-ile	Maximum
New-Order	0.15s	0.26s	5.32s
Payment	0.14s	0.25s	5.39s
Order-Status	0.15s	0.26s	5.01s
Delivery (interactive portion)	0.13s	0.18s	2.12s
Delivery (deferred portion)	0.03s	0.07s	2.43s
Stock-Level	0.20s	0.33s	2.16s
Menu	0.13s	0.18s	2.91s

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	18.02s	18.41s
Payment	3.00	3.02s	3.41s
Order Status	2.00	2.02s	2.4s
Interactive Delivery	2.00	2.02s	2.4s
Stock Level	2.00	2.02s	2.41s

Table 6: Transaction Think Times

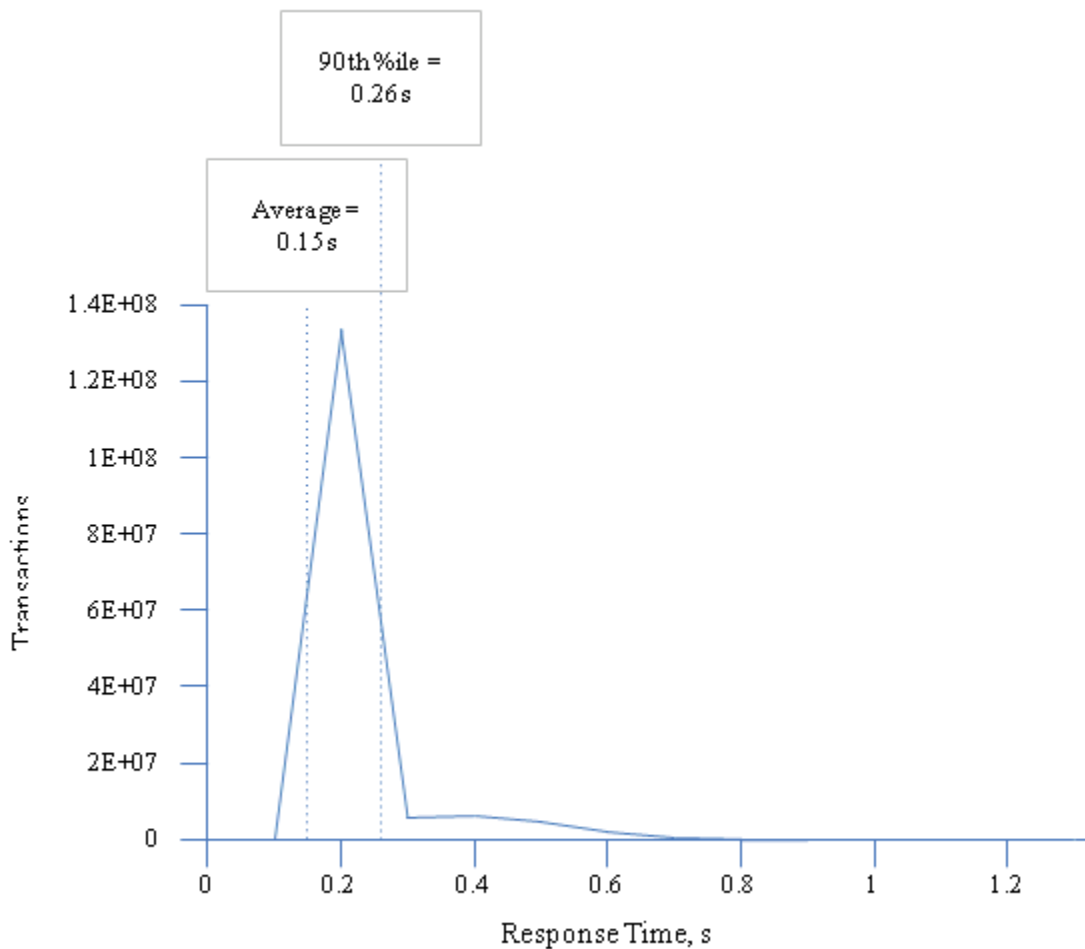
Think Times	Minimum	Average	Maximum
New Order	0	12.06s	120.53s
Payment	0	12.06s	120.53s
Order Status	0	10.05s	100.52s
Interactive Delivery	0	5.07s	50.52s
Stock Level	0	5.07s	50.52s

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

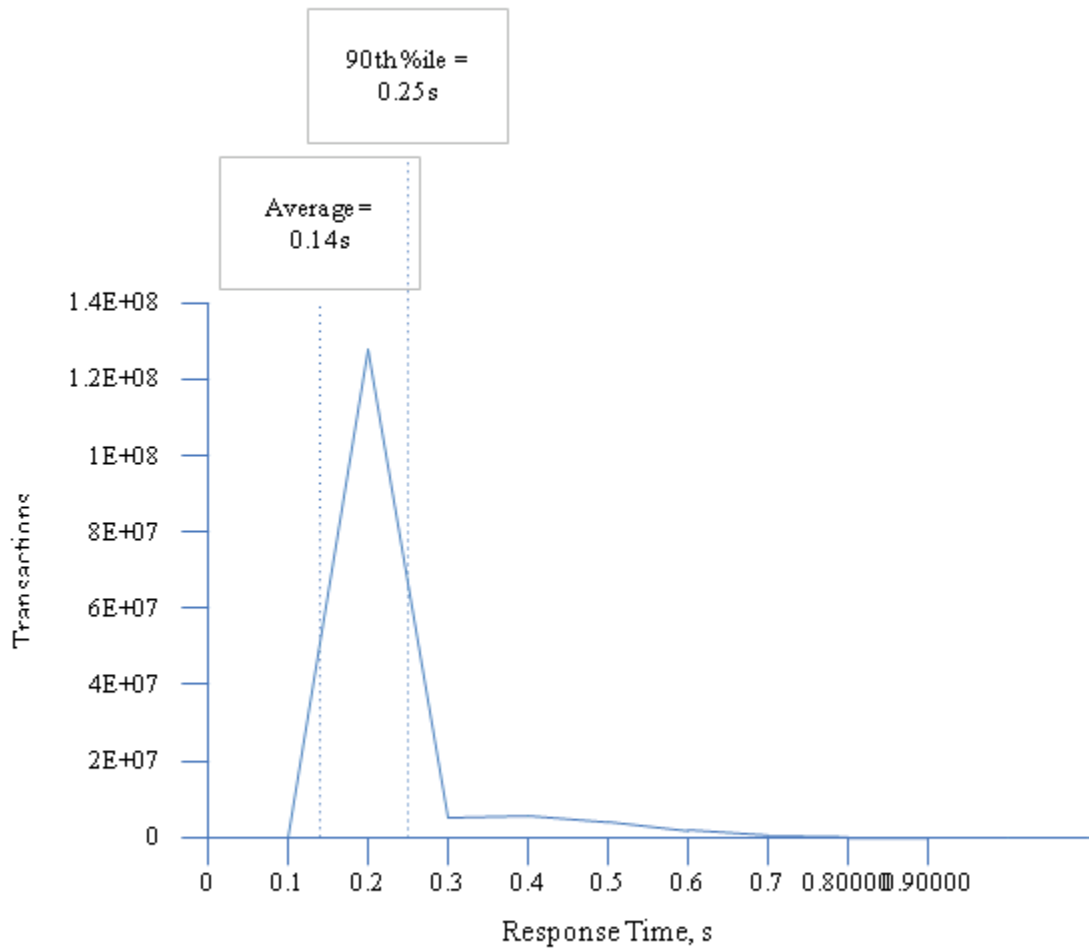
Figure 3: New Order Response Time Distribution



Response time frequency distribution for New Order transaction

5.4.2 Payment Response Time Distribution

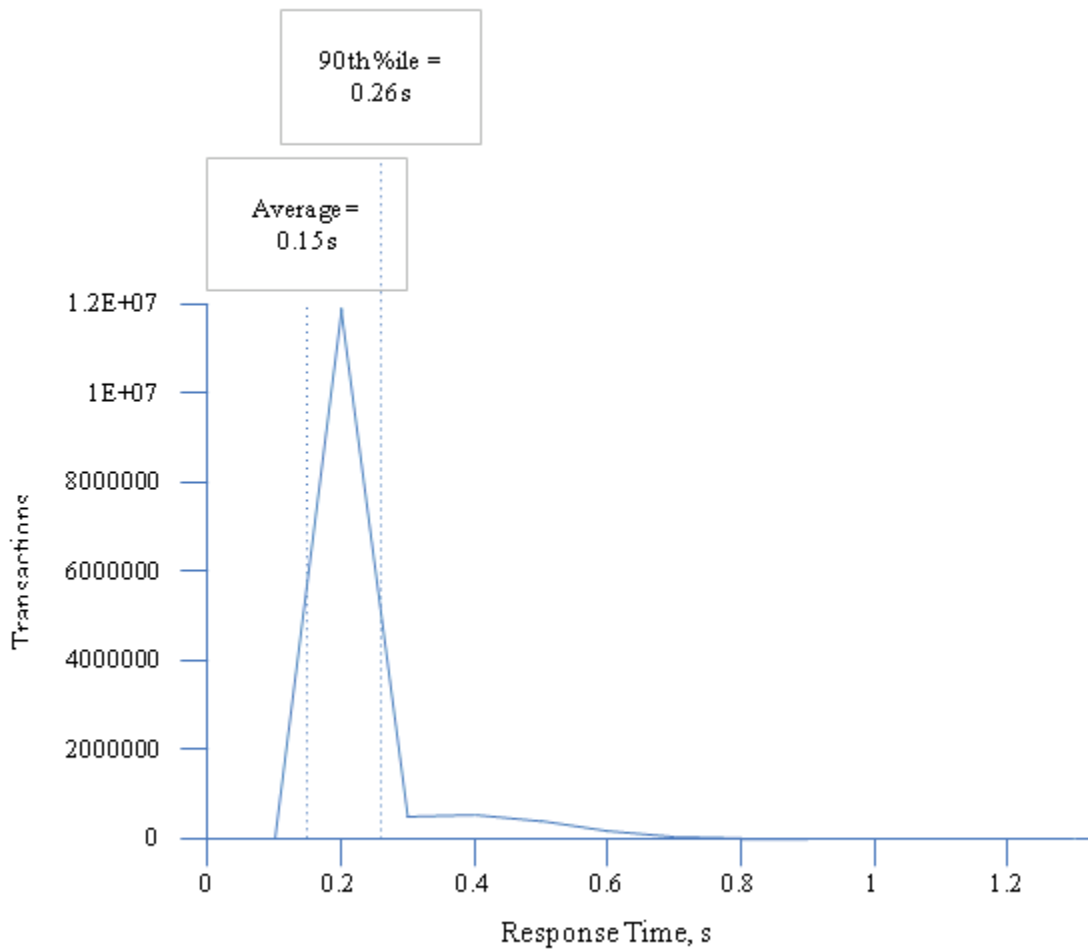
Figure 4: Payment Response Time Distribution



Response time frequency distribution for Payment transaction

5.4.3 Order Status Response Time

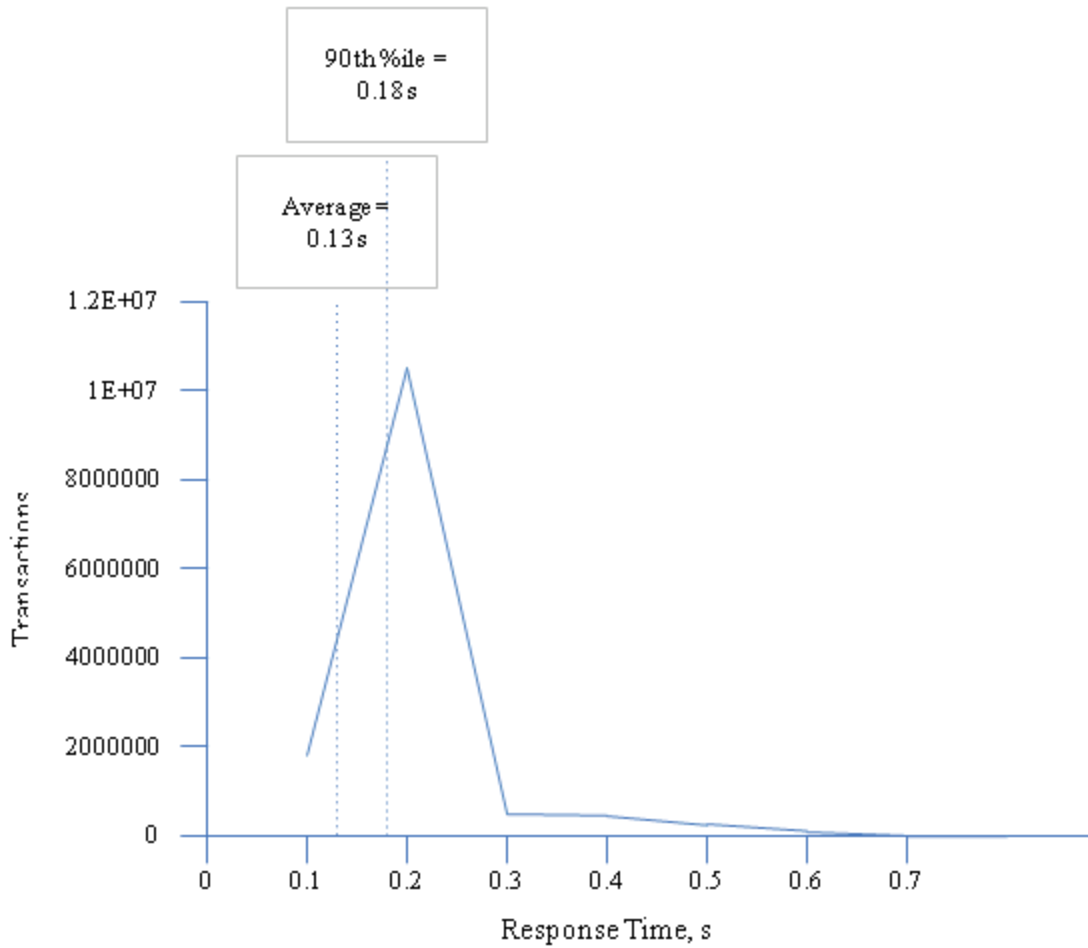
Figure 5: Order Status Response Time Distribution



Response time frequency distribution for Order Status transaction

5.4.4 Delivery Response Time Distribution

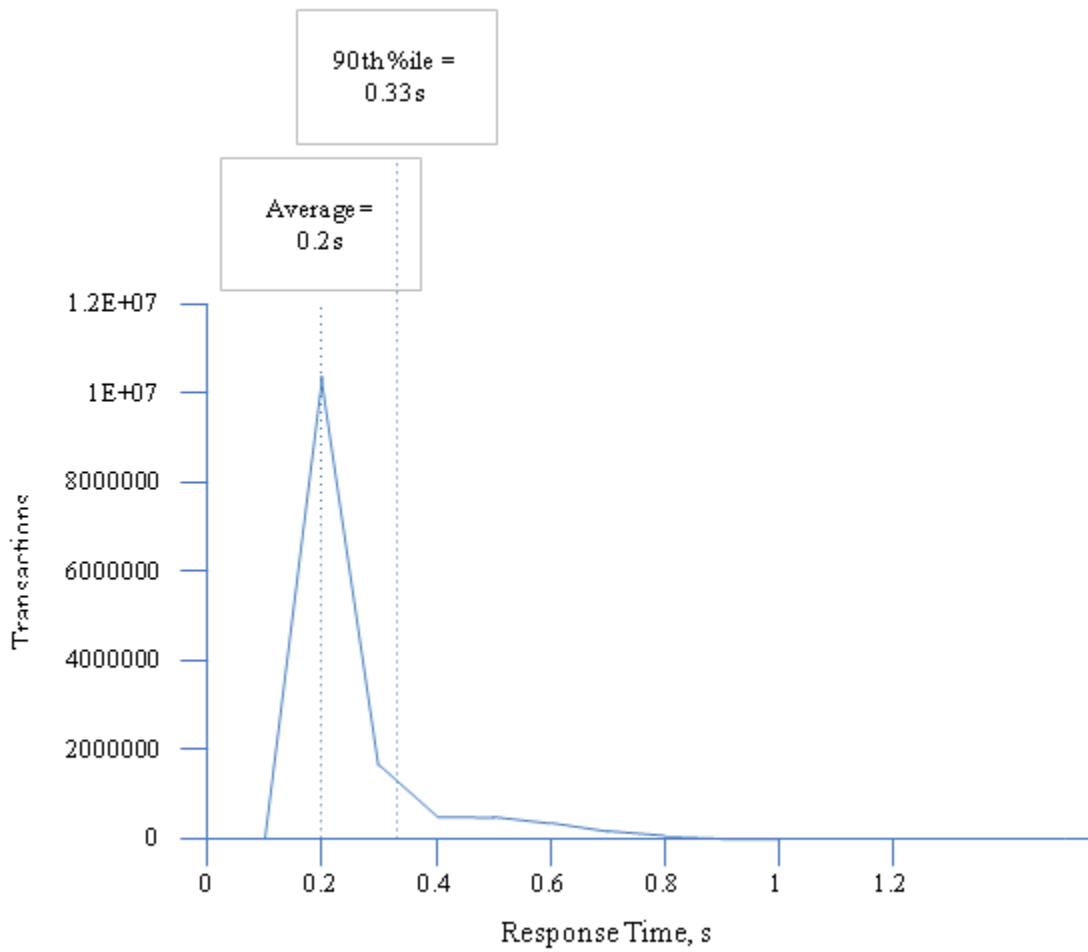
Figure 6: Delivery Response Time Distribution



Response time frequency distribution for Delivery transaction

5.4.5 Stock Level Response Time

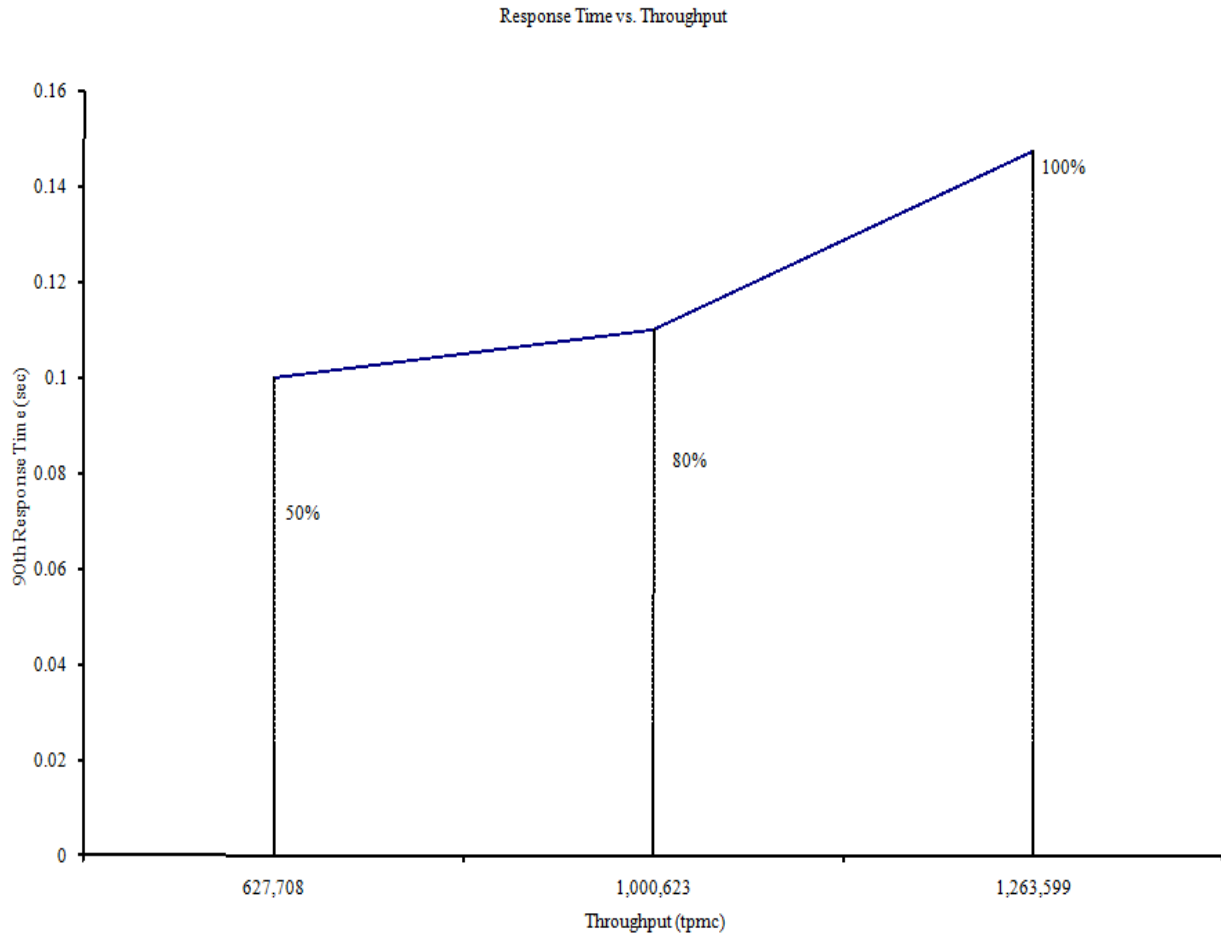
Figure 7: Stock Level Response Time Distribution



Response time frequency distribution for Stock Level transaction

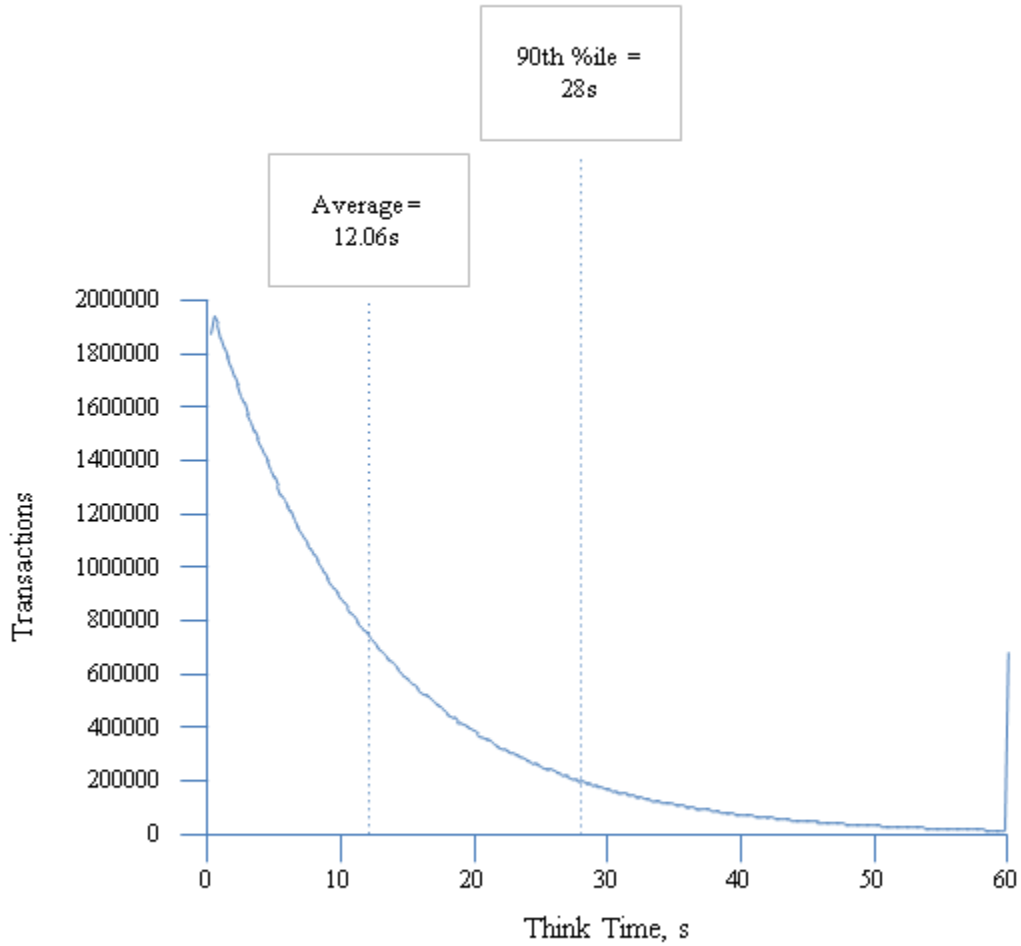
5.4.6 Response Time Versus Throughput

Figure 8: New Order Response Time Distribution



5.4.7 New Order Think Time Distribution

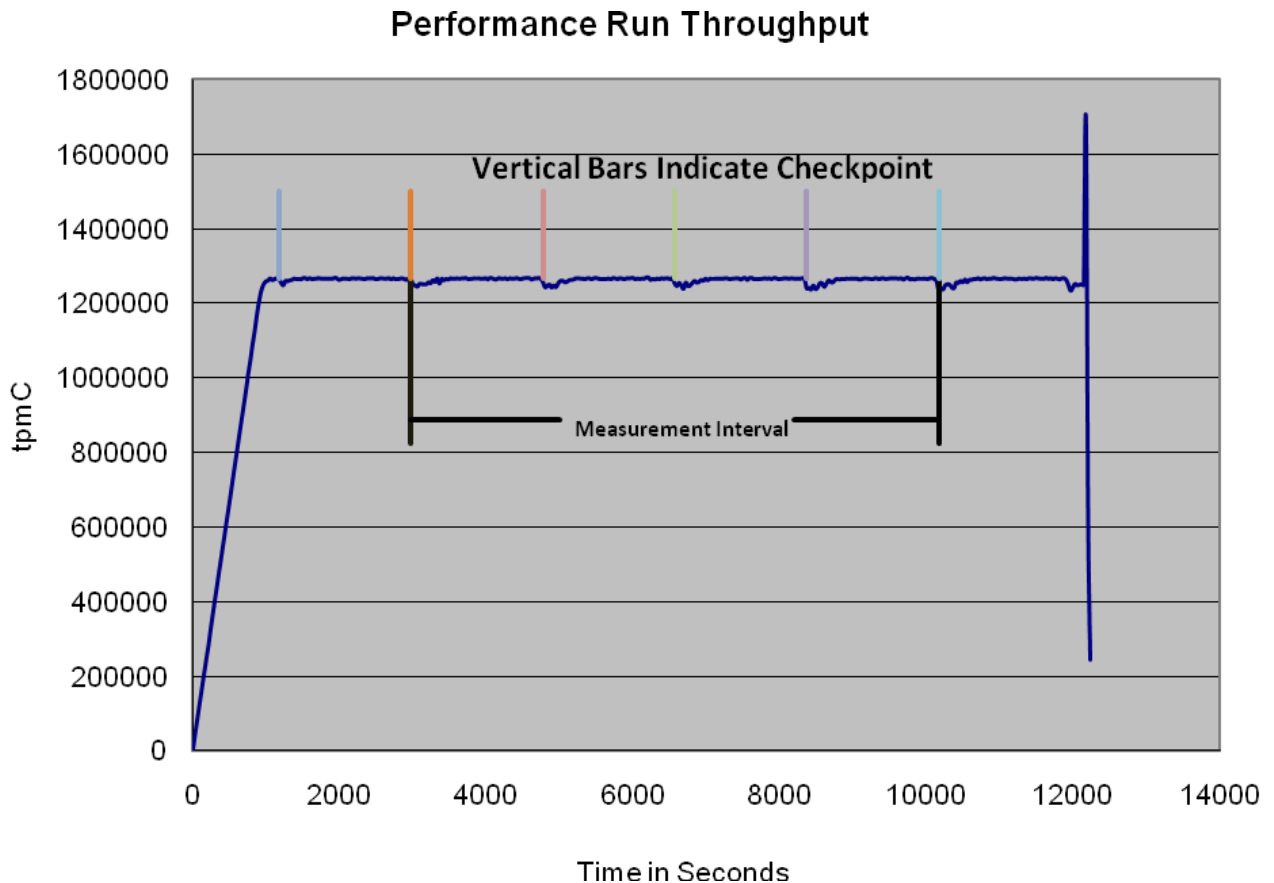
Figure 9: New Order Think Time Distribution



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 1750" syntax in Microsoft SQL Server 2005 Enterprise X64 Edition SP3 was used to force the checkpoints to an interval of 29 minutes, 10 seconds. 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 X64 Edition SP3 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 X64 Edition SP3 wrote a special record to the recovery log to indicate that all disk operations had been satisfied to this point. The positioning of the checkpoint was verified to be clear of the guard zones.

5.7 Measurement Period Duration

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

The measurement interval was 120 minutes.

5.8 Regulation of Transaction Mix

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

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

5.9 Transaction Mix

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

Table 7: Transaction Mix

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

5.10 Transaction Statistics

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

Table 1 contains the required items.

5.11 Checkpoint Count and Location

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

The measurement interval is 120 minutes. There are 4 checkpoints within the measurement interval and 1 checkpoint before the measurement interval.

Table 8: Measurement Interval and Checkpoints

Event	From	To
Measured Interval	20:09:16	22:09:16
Checkpoint	19:39:26	20:08:36
Checkpoint	20:09:16	20:38:26
Checkpoint	20:39:06	21:08:16
Checkpoint	21:08:56	21:38:06
Checkpoint	21:38:46	22:07:56
Checkpoint	22:08:36	22:37:46

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, another Gigabit network was used between the clients and the database server.

6.5 Operator Intervention

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

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

Chapter 7 Pricing

7.1 System Pricing

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

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

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

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

7.2 General Availability, Throughput and Price Performance

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

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

Table 9: Throughput, Price Performance and Availability

Maximum qualified throughput:	1,263,599 tpmC
Price per tpmC:	\$0.51 USD per tpmC
Availability:	May 23, 2011

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:

- 4 Microsoft SQL Server 2005 Enterprise X64 Edition SP3 per-processor licenses.
- Microsoft Windows Server 2008 R2 Enterprise Edition for X64-Based Systems
- 8 Microsoft Windows Server 2008 Standard Licenses

- 3 year support for hardware components

7.5 Testing

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
10955 Tantau Avenue
Cupertino, CA 95014-0770 USA

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



May 19, 2011

Mr. Eric Deehr
Performance Engineer
Hewlett-Packard Company
14475 24th Street NE
Bellevue, WA 98007

I have verified on-site the TPC Benchmark™ C for the following configuration:

Platform: HP ProLiant BL685c G7
Database Manager: Microsoft SQL Server 2005 Enterprise X64 Edition
Operating System: Microsoft Windows Server 2008 R2 Enterprise Edition
Transaction Monitor: Microsoft COM+

System Under Test:				
CPU's	Memory	Disks (total)	90% Response	TpmC
4 AMD 12 core @ 2200 Mhz	Main: 512 GB	6 SSD @ 1.56 TB 20 @ 500 GB 2 @ 72 GB	0.26	1,263,599
Clients: 8 BL460C G6				
2 Xeon 4 core @ 2.27 Ghz	8 GB	2 @ 146 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.
- The database was properly scaled with 100,000 warehouses, 99,921 of which were active during the measured interval.
- 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 tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 days space calculation was verified.
- The steady state portion of the test was 120 minutes.
- There was one complete checkpoint in steady state before the measured interval.
- There were 4 checkpoints started and completed inside the measured interval.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: None

Sincerely,



Lorna Livingtree
TPC Certified Auditor



Appendix A Source Code

Isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

Isapi_dll/src/tpcc.h

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

//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE        101
#define _APS_NEXT_COMMAND_VALUE       101
        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
#define PAYMENT_FORM form id

        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;
} TERM; //pointer to allocated client data
typedef TERM *PTERM;

//pointer to terminal
structure type

enum WEBERROR
{
    NO_ERR,

    ERR_COMMAND_UNDEFINED,
    ERR_D_ID_INVALID,

    ERR_DELIVERY_CARRIER_ID_RAN
GE,

    ERR_DELIVERY_CARRIER_INVALID,

    ERR_DELIVERY_MISSING_OCD_KE
Y,

    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILL_FORMED,
    ERR_INVALID_SYNC_CONNECTION
,

    ERR_INVALID_TERMID,
    ERR_LOADDLL_FAILED,
    ERR_MAX_CONNECTIONS_EXCEED
ED,

    ERR_MEM_ALLOC_FAILED,
    ERR_MISSING_REGISTRY_ENTRIES
,

    ERR_NEWORDER_CUSTOMER_INV
ALID,

    ERR_NEWORDER_CUSTOMER_KEY,
    ERR_NEWORDER_DISTRICT_INVAL
ID,

    ERR_NEWORDER_FORM_MISSING_
DID,

    ERR_NEWORDER_ITEMID_INVALID
,

    ERR_NEWORDER_ITEMID_RANGE,
    ERR_NEWORDER_ITEMID_WITHOU
T_SUPPW,

    ERR_NEWORDER_MISSING_IID_KE
Y,

    ERR_NEWORDER_MISSING_QTY_K
EY,

    ERR_NEWORDER_MISSING_SUPPW
_KEY,
```

```

ED, ERR_NEWORDER_NOITEMS_ENTER
, ERR_NEWORDER_QTY_INVALID,
, ERR_NEWORDER_QTY_RANGE,
SUPPW, ERR_NEWORDER_QTY_WITHOUT_
, 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_CID
_KEY,
, ERR_ORDERSTATUS_MISSING_DID
_KEY,
, ERR_PAYMENT_CDI_INVALID,
, ERR_PAYMENT_CID_AND_CLT,
D, ERR_PAYMENT_CUSTOMER_INVALI
,
, ERR_PAYMENT_CWI_INVALID,
, ERR_PAYMENT_DISTRICT_INVALID
,
, ERR_PAYMENT_HAM_INVALID,
, ERR_PAYMENT_HAM_RANGE,
ONG, ERR_PAYMENT_LAST_NAME_TO_L
,
, 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_KE
Y,

```

```

ERR_STOCKLEVEL_MISSING_THRE
SHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_IN
VALID,
ERR_STOCKLEVEL_THRESHOLD_RA
NGE,
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
//the delivrv executable
#define TXN_EVENT_START
2
#define TXN_EVENT_STOP
4
#define TXN_EVENT_WARNING 6
//used to record a
warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void
ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId);
void
WelcomeForm(EXTENSION_CONTROL_BLOCK
*pECB, char *szBuffer);

void SubmitCmd(EXTENSION_CONTROL_BLOCK
*pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK
*pECB, int iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK
*pECB, char *szBuffer);
void
ErrorMessage(EXTENSION_CONTROL_BLOCK
*pECB, int iError, int iErrorType, char *szMsg, int
iTermId);
void GetKeyValue(char **pQueryString, char
*pKey, char *pValue, int iMax, WEBERROR err);
int GetIntKeyValue(char **pQueryString, char
*pKey, WEBERROR NoKeyErr, WEBERROR
NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK
*pECB, int iType, int iErrorNum, int iTermId, int
iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int
iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId,
STOCK_LEVEL_DATA *pStockLevelData, BOOL
bInput, char *szForm);
void MakeNewOrderForm(int iTermId,
NEW_ORDER_DATA *pNewOrderData, BOOL
bInput, char *szForm);
void MakePaymentForm(int iTermId,
PAYMENT_DATA *pPaymentData, BOOL bInput,
char *szForm);
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData,
BOOL bInput, char *szForm);
void MakeDeliveryForm(int iTermId,
DELIVERY_DATA *pDeliveryData, BOOL bInput,
char *szForm);
void
ProcessNewOrderForm(EXTENSION_CONTROL_B
LOCK
*pECB, int iTermId, char
*szBuffer);
void
ProcessPaymentForm(EXTENSION_CONTROL_B
LOCK
*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"

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

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

#ifdef _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"
```

```
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
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
////////////////////////////////////
//
//
// 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 <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 "420"

static CRITICAL_SECTION
TermCriticalSection;

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

TYPE_CTPCC_DBLIB
*pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC
*pCTPCC_ODBC_new;

```

```

TYPE_CTPCC_TUXEDO
TYPE_CTPCC_ENCINA
TYPE_CTPCC_ODBC
TYPE_CTPCC_COM;
*pCTPCC_ENCINA_post_init;
TYPE_CTPCC_COM
*pCTPCC_COM_new;

// For deferred Delivery txns:

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

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

// configuration settings from registry
TPCCREGISTRYDATA Reg;

DWORD
dwNumDeliveryThreads = 4;
CRITICAL_SECTION
DelBuffCriticalSection;
//critical section for delivery
transactions cache
DELIVERY_TRANSACTION *pDelBuff
= NULL;
DWORD
dwDelBuffSize
= 100;
// size of circular buffer for delivery
txns
DWORD
dwDelBuffFreeCount;

// number of buffers free
DWORD
dwDelBuffBusyIndex
= 0;
// index
position of entry waiting to be delivered
DWORD
dwDelBuffFreeIndex
= 0;
// index
position of unused entry

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

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

/* FUNCTION: DllMain
*
* PURPOSE: This function is the entry point for
the DLL. This implementation is based on the
fact that
DLL_PROCESS_ATTACH is only called from the
inet service once.
*
* ARGUMENTS: HANDLE hModule
module handle

```

```

*
DWORD ul_reason_for_call
reason for call
LPVOID lpReserved
reserved for future
use
*
* RETURNS: BOOL FALSE
errors 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_TUXED
O_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_ENCIN
A_new");

```

```

        pCTPCC_ENCINA_post_init =
(TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCIN
A_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_n
ew");

        if (pCTPCC_COM_new
== NULL)

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

        // load DLL for database connection

        if ((Reg.eTxnMon == None) ||
(dwNumDeliveryThreads > 0))
        {
            if (Reg.eDB_Protocol
== DBLIB)

                {

                    strcpy(
szDllName, Reg.szPath );

                    strcat(
szDllName, "tpcc_dblib.dll");

```

```

                    hLibInstanceDb = LoadLibrary(
szDllName );

                    if
(hLibInstanceDb == NULL)

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

                    // get
function pointer to wrapper for class constructor

                    pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_
new");

                    if
(pCTPCC_DBLIB_new == NULL)

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

                    }

                    else if
(Reg.eDB_Protocol == ODBC)
                    {

                        strcpy(
szDllName, Reg.szPath );

                        strcat(
szDllName, "tpcc_odbc.dll");

                        hLibInstanceDb = LoadLibrary(
szDllName );

                        if
(hLibInstanceDb == NULL)

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

                        // get
function pointer to wrapper for class constructor

                        pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_
new");

                        if
(pCTPCC_ODBC_new == NULL)

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

```

```

    }
}

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

    //
    InitializeCriticalSection(&hConnectC
riticalSection);

    // for deferred
    delivery txns:

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

    InitializeCriticalSection(&DelBuffCritic
alSection);

    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(&DelBuffCritic
alSection);

        // Delete delivery
        delay critical section

        //
        DeleteCriticalSection(&hConnectCritic
alSection);
    }

    DeleteCriticalSection(&TermCriticalS
ection);

    if (hLibInstanceTm != NULL)
        FreeLibrary(
hLibInstanceTm );
    hLibInstanceTm = NULL;

    if (hLibInstanceDb != NULL)
        FreeLibrary(
hLibInstanceDb );
    hLibInstanceDb = NULL;

    Sleep(500);

    break;

    default:

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

        _sntprintf(szMsg,
sizeof(szMsg), "%s error, code %d: %s",
e->ErrorTypeStr(), e->ErrorNum(),
e->ErrorText());

        WriteMessageToEventLog( szMsg );
        delete e;

        TerminateExtension(0);
        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("U
nhandled 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.
 *
 *                 Release
 *                 all resources in anticipation of being unloaded.
 *
 * RETURNS:        TRUE      inet
 *                 service expected return value.
 */

BOOL WINAPI TerminateExtension( DWORD
dwFlags )
{
    if (pDeliHandles)
    {
        SetEvent( hDoneEvent
    );
        for(DWORD i=0;
i<dwNumDeliveryThreads; i++)

            WaitForSingleObject(
pDeliHandles[i], INFINITE );

        TermDeleteAll();
        return TRUE;
    }
}

/* FUNCTION: HttpExtensionProc
 *
 * PURPOSE: This function is the main entry point
 * for the TPCC DLL. The internet service
 *
 *                 calls this
 *                 function passing in the http string.

```

```

 * ARGUMENTS:
 *                 EXTENSION_CONTROL_BLOCK
 *                 *pECB      structure pointer to
 *                 passed in internet
 *
 *
 *                 service
 *                 information.
 *
 * RETURNS:        DWORD
 *                 HSE_STATUS_SUCCESS
 *
 *                 connection can be dropped if error
 *
 *                 HSE_STATUS_SUCCESS_AND_KEEP
 *                 _CONN      keep connect valid comment sent
 *
 * COMMENTS:      None
 */

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

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

#ifdef ICECAP
    StartCAP();
#endif

    try
    {
        //process http query

        ProcessQueryString(pECB, &iCmd,
&FormId, &TermId, &iSyncId);

        if (TermId != 0)
        {
            if (
TermId < 0 || TermId >= Term.iNumEntries ||
Term.pClientData[TermId].iNextFree != -1 )
            {
                // debugging...

                char szTmp[128];

                sprintf( 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
(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;
    }
}
case 2:

```

```

// new-
order selected from menu; display new-order
input form

    MakeNewOrderForm(TermId, NULL,
INPUT_FORM, szBuffer);

        break;
    case 3:
        //
payment selected from menu; display payment
input form

    MakePaymentForm(TermId, NULL,
INPUT_FORM, szBuffer);

        break;
    case 4:
        //
delivery selected from menu; display delivery
input form

    MakeDeliveryForm(TermId, NULL,
INPUT_FORM, szBuffer);

        break;
    case 5:
        // order-
status selected from menu; display order-status
input form

    MakeOrderStatusForm(TermId,
NULL, INPUT_FORM, szBuffer);

        break;
    case 6:
        // stock-
level selected from menu; display stock-level
input form

    MakeStockLevelForm(TermId,
NULL, INPUT_FORM, szBuffer);

        break;
    case 7:
        //
ExitCmd

    TermDelete(TermId);

    WelcomeForm(pECB, szBuffer);
        break;
    case 8:

    SubmitCmd(pECB, szBuffer);
        break;
    case 9:
        // menu

    MakeMainMenuForm(TermId,
Term.pClientData[TermId].iSyncId, szBuffer);
        break;
    case 10:
        //
CMD=Clear

        // resets
all connections; should only be used when no
other connections are active

    TermDeleteAll();

    TermInit();

    WelcomeForm(pECB, szBuffer);
        break;
    case 11:
        //
CMD=Stats

    StatsCmd(pECB, szBuffer);
        break;

```

```

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

#ifdef ICECAP
    StopCAP();
#endif

    lpbSize = strlen(szBuffer);
    wsprintf(szHeader1,

"Content-Type: text/html\r\n"

"Content-Length: %d\r\n"

"Connection: Keep-Alive\r\n\r\n",
lpbSize);
    strcat( szHeader1, szBuffer );

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

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

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

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

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

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

        (VOID)
DeregisterEventSource(hEventSource);
    }
}

```

```

}

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

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

    DELIVERY_TRANSACTION
delivery;
PDELIVERY_DATA

    pDeliveryData;
TXN_RECORD_TPCC_DELIV_DEF
txnDeliRec;

    DWORD
index;

    HANDLE
handles[2];

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

    assert(txnDeliLog != NULL);

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

                //
EnterCriticalSection(&hConnectCriti-
calSection);

                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("U
            nhandled 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(&DelBuffCritical
            Section);

            delivery =
            *(pDelBuff+dwDelBuffBusyIndex);

            dwDelBuffFreeCount++;
            dwDelBuffBusyIndex++;

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

                dwDelBuffBusyIndex
                = 0;

            LeaveCriticalSection(&DelBuffCritical
            Section);

            pDeliveryData->w_id =
            delivery.w_id;

            pDeliveryData->o_carrier_id =
            delivery.o_carrier_id;

            txnDeliRec.w_id = pDeliveryData-
            >w_id;

            txnDeliRec.o_carrier_id =
            pDeliveryData->o_carrier_id;

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

            GetLocalTime( &trans_start );

            pTxn->Delivery();

            GetLocalTime( &trans_end );

            //log txn

            txnDeliRec.TxnStatus =
            ERR_SUCCESS;

            for (int i=0; i<10; i++)

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

```

```

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

            if (txnDeliLog != NULL)

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

                e->ErrorTypeStr(),
                e->ErrorNum(), e->ErrorText() );

                WriteMessageToEventLog( szTmp );

                // log
                the error txn

                txnDeliRec.TxnStatus = e-
                >ErrorType();

                if
                (txnDeliLog != NULL)

                    txnDeliLog-
                    >WriteToLog(&txnDeliRec);

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

                WriteMessageToEventLog(TEXT("U
                nhandled 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(&DelBuffCritical
Section);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;

        (pDelBuff+dwDelBuffFreeIndex)-
>w_id
                = w_id;

        (pDelBuff+dwDelBuffFreeIndex)-
>o_carrier_id= o_carrier_id;

        GetLocalTime(&(pDelBuff+dwDelBu
ffFreeIndex)->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(&DelBuffCritical
Section);

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

    return bError;
}

/* FUNCTION: ProcessQueryString
*
* PURPOSE: This function extracts the relevent
information out of the http command passed in
from
*
                the
browser.
*
* COMMENTS:          If this is the initial
connection i.e. client is at welcome screen then

```

```

*
                there will not be a terminal id or
current form id. If this is the case
                then the pTermid and pFormid
return values are undefined.
*/

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

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

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

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

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

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

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

/* FUNCTION: void WelcomeForm
*
*/

```

```

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

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

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

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

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

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

        "</PRE></font>"

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

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

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

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

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

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

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

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

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

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

        "Max Connections
= <B>%d</B><BR>"

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

        "Max Pending
Deliveries = <B>%d</B><BR>"

```

```

szTxnMonNames[Reg.eTxnMon],
szDBNames[Reg.eDB_Protocol],

    Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

        "</PRE></font>"

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

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

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

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

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

```

```

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

"</PRE></font>"

, Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

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

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

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

    if (Reg.eTxnMon == None)
    {
        // parse Server name

```

```

GetKeyValue(&ptr,
"db_server", szServer, sizeof(szServer),
ERR_NO_SERVER_SPECIFIED);
// parse User name
GetKeyValue(&ptr,
"db_user", szUser, sizeof(szUser), NO_ERR);
// parse Password
GetKeyValue(&ptr,
"db_passwd", szPassword, sizeof(szPassword),
NO_ERR);
// parse Database
name
GetKeyValue(&ptr,
"db_name", szDatabase, sizeof(szDatabase),
NO_ERR);
}

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

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

iNewTerm = TermAdd();

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

try
{
    if (Reg.eTxnMon ==
TUXEDO)

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

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

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

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

        szDatabase, Reg.szSPPrefix,

        Reg.bCallNoDuplicatesNewOrder );

```

```

else if
(Reg.eDB_Protocol == DBLIB)

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

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

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

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

    EnterCriticalSection(&TermCriticalSe
ction);

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

            iTotal++;
    }

    LeaveCriticalSection(&TermCriticalS
ection);

    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 *CWEBCNT_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_RAN
GE,
    "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_KE
Y,
    "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_EXCEED
ED,
    "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_INV
ALID,
    "New
Order customer id invalid data type, range = 1 to
3000."
},
{
    ERR_NEWORDER_CUSTOMER_KEY,

    "New Order missing Customer key
\"CID*\"."
},
{
    ERR_NEWORDER_DISTRICT_INVAL
ID,
    "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_WITHOU
T_SUPPW,
    "New Order Item_Id
field entered without a corresponding Supp_W."
},
{
    ERR_NEWORDER_MISSING_IID_KE
Y,
    "New
Order missing Item Id key \"IID*\"."
},
{
    ERR_NEWORDER_MISSING_QTY_K
EY,
    "New
Order Missing Qty key \"Qty##*\"."
},
{
    ERR_NEWORDER_MISSING_SUPPW
_KEY,
    "New
Order missing Supp_W key \"SP##*\"."
},
{
    ERR_NEWORDER_NOITEMS_ENTER
ED,
    "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
    D, "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_L
    ONG, "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
    Y, "Payment missing Amount key
    \"HAM*\"."
    },
    ERR_STOCKLEVEL_MISSING_THRE
    SHOLD_KEY, "Stock Level; missing Threshold key
    \"TT*\"."
    },
    ERR_STOCKLEVEL_THRESHOLD_IN
    VALID, "Stock Level;
    Threshold value must be in the range = 1 - 99."
    },
    ERR_STOCKLEVEL_THRESHOLD_RA
    NGE, "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)
    wprintf(
szTmp+strlen(szTmp), " Error=%d",
m_SystemErr );

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

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

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

```

```

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

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

*pQueryString = ptr;
return;

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

/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http
formatted string for a specific key value.
*
* ARGUMENTS:      char
                  *pQueryString
                  http string from client browser
*
* char
* *pKey
key value to look for
*
* WEBERROR
* NoKeyErr      error
value to throw if key not found
*
* WEBERROR
* NotIntErr     error
value to throw if value not numeric
*
* RETURNS:      integer
*
* ERROR:      if (the pKey value is
not found) then
*
if (NoKeyErr !=
NO_ERR)
    throw
CWEBCLNT_ERR(err)
*
else
return 0
*
else if (non-numeric char found)
then
*
if (NotIntErr !=
NO_ERR) then
    throw
CWEBCLNT_ERR(err)
*
else

```

```

*
* return 0
*
* COMMENTS:      http keys are
formatted either KEY=value& or KEY=value\0.
This DLL formats
*
TPC-C input fields in such a manner
that the keys can be extracted in the
*
above manner.
*/

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

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

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

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

    *pQueryString = ptr;
    return atoi(ptr0);

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

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

void TermInit(void)
{
    EnterCriticalSection(&TermCriticalSe
ction);

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

```

```

        Term.pClientData = NULL;
        Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries *
sizeof(CLIENTDATA));
        if (Term.pClientData == NULL)
        {
            LeaveCriticalSection(&TermCriticalS
ection);
            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(&TermCriticalS
ection);
    }

/* 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(&TermCriticalSe
ction);

    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(&TermCriticalS
ection);
}

```

```

/* 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(&TermCriticalSe
ction);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm =
Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;

        Term.pClientData[iNewTerm].iNext
Free = -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(&TermCriticalS
ection);
            throw
new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
        }

        Term.pClientData[iNewTerm].iTickC
ount = GetTickCount();
        Term.pClientData[iNewTerm].iSyncI
d = Term.iMasterSyncId++;
    }
}

```

```

        Term.pClientData[iNewTerm].pTxn
= NULL;
        LeaveCriticalSection(&TermCriticalS
ection);
        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(&TermCriticalSe
ction);

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

        LeaveCriticalSection(&TermCriticalS
ection);
    }
}

/* FUNCTION: MakeErrorForm
*/

void ErrorForm(EXTENSION_CONTROL_BLOCK
*pECB, int iType, int iErrorNum, int iTermId, int
iSyncId, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,

        "<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
        "<FORM
ACTION=\"%tcc.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)
{
    wsprintf(szForm,

        "<HTML><HEAD><TITLE>TPC-C
Main Menu</TITLE></HEAD><BODY>"
        "Select Desired
Transaction.<BR><HR>"
        "<FORM
ACTION='\"tpcc.dll\" METHOD='\"GET\">"
        "<INPUT
TYPE='hidden' NAME='STATUSID'
VALUE='\"0\">"
        "<INPUT
TYPE='hidden' NAME='ERROR'
VALUE='\"0\">"
        "<INPUT
TYPE='hidden' NAME='FORMID'
VALUE='\"%d\">"
        "<INPUT
TYPE='hidden' NAME='TERMINID'
VALUE='\"%d\">"
        "<INPUT
TYPE='hidden' NAME='SYNCD'
VALUE='\"%d\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..NewOrder..\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Payment..\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Delivery..\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Order-Status..\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Stock-Level..\">"
        "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Exit..\">"

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

```

```

        , MAIN_MENU_FORM,
iTermId, iSyncId);

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

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

    c = wsprintf(szForm,

        "<HTML><HEAD><TITLE>TPC-C
Stock Level</TITLE></HEAD><FORM
ACTION='\"tpcc.dll\" METHOD='\"GET\">"
        "<INPUT
TYPE='hidden' NAME='STATUSID'
VALUE='\"0\">"
        "<INPUT
TYPE='hidden' NAME='ERROR'
VALUE='\"0\">"
        "<INPUT
TYPE='hidden' NAME='FORMID'
VALUE='\"%d\">"
        "<INPUT
TYPE='hidden' NAME='TERMINID'
VALUE='\"%d\">"
        "<INPUT
TYPE='hidden' NAME='SYNCD'
VALUE='\"%d\">"
        "<PRE><font
face='\"Courier\"> Stock-
Level<BR>"
        "Warehouse: %6.6d
District: %2.2d<BR><BR>",
        STOCK_LEVEL_FORM,
iTermId, Term.pClientData[iTermId].iSyncId,

        Term.pClientData[iTermId].w_id,
Term.pClientData[iTermId].d_id);

    if ( bInput )
    {
        strcpy(szForm+c,

            "Stock
Level Threshold: <INPUT NAME='TT*'
SIZE=2><BR><BR>"
            "low
stock: </font><BR><BR><BR><BR>
<BR><BR><BR><BR><BR><BR>"
            "<BR>
<BR><BR><BR><BR><BR>
<BR></PRE><HR>"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"Process\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"Menu\">"

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

```

```

        else
            wsprintf(szForm+c,
"Stock
Level Threshold: %2.2d<BR><BR>"
            "low
stock: %3.3d</font><BR><BR><BR><BR>
<BR><BR><BR><BR><BR><BR>"
            "<BR>
<BR><BR><BR><BR><BR><BR>
<BR></PRE><HR>"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..NewOrder..\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Payment..\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Delivery..\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Order-Status..\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Stock-Level..\">"
            "<INPUT
TYPE='submit' NAME='CMD'
VALUE='\"..Exit..\">"

            "</FORM></HTML>"

        pStockLevelData->threshold, pStockLevelData-
>low_stock);
    }

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

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

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

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

    c = wsprintf(szForm,

        "<HTML><HEAD><TITLE>TPC-C
New Order</TITLE></HEAD><BODY>"
        "<FORM
ACTION='\"tpcc.dll\" METHOD='\"GET\">"
        "<INPUT
TYPE='hidden' NAME='STATUSID'
VALUE='\"%d\">"

```



```

" <INPUT
TYPE="hidden" NAME="ERROR"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="FORMID"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="TERMINID"
VALUE="0">"
" <INPUT
TYPE="hidden" NAME="SYNCID"
VALUE="0">"
" <PRE><font
face="Courier"> New
Order<BR>"
, bValid ? 0 :
ERR_BAD_ITEM_ID, NEW_ORDER_FORM,
iTermId, Term.pClientData[iTermId].iSyncId);
if ( bInput )
{
c +=
sprintf(szForm+c, "Warehouse: %6.6d ",
Term.pClientData[iTermId].w_id );
strcpy( szForm+c,
"District:
<INPUT NAME="DID*" SIZE=1>
Date: <BR>"
"Customer: <INPUT
NAME="CID*" SIZE=4> Name:
Credit: %Disc:<BR>"
"Order
Number: Number of Lines: W_tax:
D_tax: <BR> <BR>"
"
Supp_W Item_Id Item Name Qty
Stock B/G Price Amount<BR>"
"
<INPUT NAME="SP00*" SIZE=4> <INPUT
NAME="IID00*" SIZE=6>
<INPUT NAME="Qty00*" SIZE=1><BR>"
"
<INPUT NAME="SP01*" SIZE=4> <INPUT
NAME="IID01*" SIZE=6>
<INPUT NAME="Qty01*" SIZE=1><BR>"
"
<INPUT NAME="SP02*" SIZE=4> <INPUT
NAME="IID02*" SIZE=6>
<INPUT NAME="Qty02*" SIZE=1><BR>"
"
<INPUT NAME="SP03*" SIZE=4> <INPUT
NAME="IID03*" SIZE=6>
<INPUT NAME="Qty03*" SIZE=1><BR>"
"
<INPUT NAME="SP04*" SIZE=4> <INPUT
NAME="IID04*" SIZE=6>
<INPUT NAME="Qty04*" SIZE=1><BR>"
"
<INPUT NAME="SP05*" SIZE=4> <INPUT
NAME="IID05*" SIZE=6>
<INPUT NAME="Qty05*" SIZE=1><BR>"
"
<INPUT NAME="SP06*" SIZE=4> <INPUT
NAME="IID06*" SIZE=6>
<INPUT NAME="Qty06*" SIZE=1><BR>"
"
<INPUT NAME="SP07*" SIZE=4> <INPUT
NAME="IID07*" SIZE=6>
<INPUT NAME="Qty07*" SIZE=1><BR>"
"
<INPUT NAME="SP08*" SIZE=4> <INPUT
NAME="IID08*" SIZE=6>
<INPUT NAME="Qty08*" SIZE=1><BR>"

```

```

"
<INPUT NAME="SP09*" SIZE=4> <INPUT
NAME="IID09*" SIZE=6>
<INPUT NAME="Qty09*" SIZE=1><BR>"
<INPUT NAME="SP10*" SIZE=4> <INPUT
NAME="IID10*" SIZE=6>
<INPUT NAME="Qty10*" SIZE=1><BR>"
"
<INPUT NAME="SP11*" SIZE=4> <INPUT
NAME="IID11*" SIZE=6>
<INPUT NAME="Qty11*" SIZE=1><BR>"
"
<INPUT NAME="SP12*" SIZE=4> <INPUT
NAME="IID12*" SIZE=6>
<INPUT NAME="Qty12*" SIZE=1><BR>"
"
<INPUT NAME="SP13*" SIZE=4> <INPUT
NAME="IID13*" SIZE=6>
<INPUT NAME="Qty13*" SIZE=1><BR>"
"
<INPUT NAME="SP14*" SIZE=4> <INPUT
NAME="IID14*" SIZE=6>
<INPUT NAME="Qty14*" SIZE=1><BR>"
"
"Execution Status:
Total:<BR>"
" </font></PRE><HR>"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Process">"
" <INPUT
TYPE="submit" NAME="CMD"
VALUE="Menu">"
" </FORM></HTML>"
}
else
{
c +=
sprintf(szForm+c, "Warehouse: %6.6d
District: %2.2d Date: ",
pNewOrderData->w_id,
pNewOrderData->d_id);
if ( bValid )
{
c +=
sprintf(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>
"
"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 %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 );
        else
            strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");
            strcat(szForm,
                " <BR></font></PRE><HR>"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Payment..\">"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Delivery..\">"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Order-Status..\">"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Stock-Level..\">"
                " <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..Exit..\">"
                " </BODY></FORM></HTML>");
        }
}

```

```

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

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

    c = 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=\"TERMID\"
VALUE=\"%d\">"
        " <INPUT
TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
        " <PRE><font
face=\"Courier\">    Order-
Status<BR>"
        " ,
        "Warehouse: %6.6d
        ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);
    if ( bInput )
    {
        strcpy(szForm+c,
            "District:
<INPUT NAME=\"DID\" SIZE=1><BR>"
            "Customer: <INPUT
NAME=\"CID\" SIZE=4> Name:
<INPUT NAME=\"CLT\" SIZE=23><BR>"
            "Cust-
Balance:<BR> <BR>"
            "Order-
Number:    Entry-Date:
Carrier-Number:<BR>"
            "Supply-
W Item-Id Qty Amount Delivery-
Date<BR> <BR> <BR> <BR> <BR>"
            " <BR>
<BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR></font></PRE>"
            " <HR><INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\"><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"

```

```

    }</BODY></FORM></HTML>");
    else
    {
        c +=
        sprintf(szForm+c,
            "District:
%2.2d<BR>"
            "Customer: %4.4d Name: %-16s
%-2s %-16s<BR>",
            pOrderStatusData->d_id,
pOrderStatusData->c_id,
            pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData-
>c_last);
        c +=
        sprintf(szForm+c, "Cust-Balance: $%9.2f<BR>
<BR>",
            pOrderStatusData->c_balance);
        c +=
        sprintf(szForm+c,
            "Order-
Number: %8.8d Entry-Date: %2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d Carrier-Number:
%2.2d<BR>"
            "Supply-
W Item-Id Qty Amount Delivery-
Date<BR>",
            pOrderStatusData->o_id,
            pOrderStatusData->o_entry_d.day,
            pOrderStatusData-
>o_entry_d.month,
            pOrderStatusData->o_entry_d.year,
            pOrderStatusData-
>o_entry_d.hour,
            pOrderStatusData-
>o_entry_d.minute,
            pOrderStatusData-
>o_entry_d.second,
            pOrderStatusData->o_carrier_id);
        for(i=0; i<
pOrderStatusData->o_ol_cnt; i++)
        {
            c +=
            sprintf(szForm+c, " %6.6d %6.6d %2.2d
    $%8.2f %2.2d-%2.2d-%4.4d<BR>",
                pOrderStatusData-
>OL[i].ol_supply_w_id,
                pOrderStatusData->OL[i].ol_i_id,
                pOrderStatusData-
>OL[i].ol_quantity,
                pOrderStatusData-
>OL[i].ol_amount,
                pOrderStatusData-
>OL[i].ol_delivery_d.day,

```

```

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

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

    strcpy(szForm+c,

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

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

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

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

    c = sprintf(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=\"TERMINID\"
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
        {
            sprintf( 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_B
LOCK *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 inetrv.
*
int
iTermId client browser
terminal id
*
*/

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

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

```

```

        pPayment->w_id =
Term.pClientData[iTermId].w_id;
        GetPaymentData(pECB-
->lpszQueryString, pPayment);

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

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

/* FUNCTION: ProcessOrderStatusForm
 *
 * PURPOSE: This function gets and validates the
input data from the Order Status
 *
 * ARGUMENTS:
        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
 *
 * ARGUMENTS:
        form
filling in
the required input variables. It then calls the
PostDeliveryInfo

```

```

 *
 * ARGUMENTS:
        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_BL
OCK *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

```

```

 *
 * ARGUMENTS:
        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_B
LOCK *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_NOITEMS_ENTERED );

                    pNewOrderData->o_ol_cnt = items;
                }

/* FUNCTION: GetPaymentData
*
* PURPOSE: This function extracts and validates
the payment form data from an http command
string.
*
* ARGUMENTS:         LPSTR
                    lpszQueryString
                    client browser http
command string
*
                    PAYMENT_DATA
                    *pPaymentData
                    pointer to payment data structure
*/

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

```

```

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

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

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

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

                _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;

                // 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_jd =
GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp,
sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
        {
            // customer id is
blank, so last name must be entered
            pOrderStatusData-
>c_id = 0;

            GetKeyValue(&ptr,
"CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
            if ( szTmp[0] == 0 )
                throw

new CWEBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

            _strupr( szTmp );
            if ( strlen(szTmp) >
LAST_NAME_LEN )
                throw

new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CLT_RANGE );

            strcpy(pOrderStatusData->c_last,
szTmp);
            // pad with spaces so
that the client layer doesn't have to do it
            // before passing
parameters to stored procedure

```

```

        iLen =
strlen(pOrderStatusData->c_last);
        memset(pOrderStatusData->c_last
+ iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData-
>c_last[LAST_NAME_LEN] = 0;
        }
        else
        {
            // parse customer id
and verify that last name was NOT entered
            if ( !IsNumeric(szTmp)
)
                throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
            pOrderStatusData-
>c_id = atoi(szTmp);

            GetKeyValue(&ptr,
"CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
            if ( szTmp[0] != 0 )
                throw

new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
        }
}

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE: This function determines if a string
is numeric. It fails if any characters other
than
numeric and null terminator are present.
*
* ARGUMENTS:      char
                  *ptr      pointer
to string to check.
*
* RETURNS:        BOOL      FALSE
                  if string is not all numeric
*
                  TRUE
                  if string contains only numeric
characters i.e. '0' - '9'
*/
BOOL IsNumeric(char *ptr)
{
    if ( *ptr == 0 )
        return FALSE;

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

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

```

```

*
*
* TRUE
*/
    if string is OK
BOOL IsDecimal(char *ptr)
{
    char *dotptr;
    BOOL bValid;

    if ( *ptr == 0 )
        return FALSE;

    // find decimal point
    dotptr = strchr( ptr, '.' );
    if (dotptr == NULL)
        // no decimal point, so
just check for numeric
        return IsNumeric(ptr);
    *dotptr = 0; // temporarily replace
decimal with a terminator

    if ( *ptr != 0 )
        bValid =
IsNumeric(ptr);
    // string starts with decimal point
    else if (*(dotptr+1) == 0)
        return FALSE; //
nothing but a decimal point is bad
    else
        bValid = TRUE;

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

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

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

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

```

```

#include <io.h>
#include <assert.h>

#include <sqltypes.h>

#ifdef ICECAP
#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: DllMain
*
* PURPOSE: This function is the entry point for
the DLL. This implementation is based on the

```

```

* fact that
DLL_PROCESS_ATTACH is only called from the
ifnet service once.
* ARGUMENTS: HANDLE hModule
module handle
*
DWORD ul_reason_for_call
reason for call
*
LPVOID lpReserved
reserved for future
use
*
* RETURNS: BOOL FALSE
errors 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(szMyComputerN
ame, &dwSize);

szMyComputerName[dwSize] = 0;
}
DisableThreadLibraryCalls((HMODU
LE)hModule);
InitializeCriticalSection(&TermCritica
lSection);

if ( ReadTPCCRegistrySettings(
&Reg )
throw new
CWEBCLNT_ERR(
ERR_MISSING_REGISTRY_ENTRIES );

```



```

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

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

TermInit();

// load DLL for txn monitor
if (Reg.eTxnMon == TUXEDO)
{
    strcpy( szDllName,
Reg.szPath );

    strcat( szDllName,
"tpcc_tuxedo.dll");

    hLibInstanceTm =
LoadLibrary( szDllName );

    if (hLibInstanceTm
== NULL)

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

// get function pointer
to wrapper for class constructor

    pCTPCC_TUXEDO_new =
(TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXED
O_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_ENCIN
A_new");

    pCTPCC_ENCINA_post_init =
(TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCIN
A_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_n
ew");

    if (pCTPCC_COM_new
== NULL)

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

    }

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

```

```

        strcpy(
szDllName, Reg.szPath );

        strcat(
szDllName, "tpcc_dblib.dll");

        hLibInstanceDb = LoadLibrary(
szDllName );

        if
(hLibInstanceDb == NULL)

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

// get
function pointer to wrapper for class constructor

        pCTPCC_DBLIB_new =
(TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_
new");

        if
(pCTPCC_DBLIB_new == NULL)

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

        }

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

            strcat(
szDllName, "tpcc_odbc.dll");

            hLibInstanceDb = LoadLibrary(
szDllName );

            if
(hLibInstanceDb == NULL)

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

// get
function pointer to wrapper for class constructor

            pCTPCC_ODBC_new =
(TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_
new");

            if
(pCTPCC_ODBC_new == NULL)

```

```

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

if (dwNumDeliveryThreads)
{
    // Initialize delivery
delay critical section
    //
    InitializeCriticalSection(&hConnectC
riticalSection);

    // for deferred
delivery txns:
    hDoneEvent =
CreateEvent( NULL, TRUE /* manual reset */,
FALSE /* initially not signalled */, NULL );

    InitializeCriticalSection(&DelBuffCritic
alSection);

    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(&DelBuffCritic
alSection);

    // Delete delivery
delay critical section
    //

    DeleteCriticalSection(&hConnectCritic
alSection);

    DeleteCriticalSection(&TermCriticalS
ection);

    if (hLibInstanceTm != NULL)
        FreeLibrary(
hLibInstanceTm );

    hLibInstanceTm = NULL;

    if (hLibInstanceDb != NULL)
        FreeLibrary(
hLibInstanceDb );

    hLibInstanceDb = NULL;

    Sleep(500);

    break;

    default:

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

    _sntprintf(szMsg,
sizeof(szMsg), "%s error, code %d: %s",
e->ErrorTypeStr(), e->ErrorNum(),
e->ErrorText());

    WriteMessageToEventLog( szMsg );
    delete e;

    TerminateExtension(0);

```

```

        return FALSE;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Un
        handled 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
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_BLOC
K *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:
        FormId )
        switch(
        {
            case WELCOME_FORM:
            case MAIN_MENU_FORM:
                break;
            case NEW_ORDER_FORM:
                ProcessNewOrderForm(pECB,
TermId, szBuffer);
                break;
            case PAYMENT_FORM:
                ProcessPaymentForm(pECB,
TermId, szBuffer);
                break;
            case DELIVERY_FORM:
                ProcessDeliveryForm(pECB, TermId,
szBuffer);
                break;
            case ORDER_STATUS_FORM:
                ProcessOrderStatusForm(pECB,
TermId, szBuffer);
                break;
        }
    }
}

```

```

        case STOCK_LEVEL_FORM:

            ProcessStockLevelForm(pECB,
                TermId, szBuffer);

                break;
            }
            break;
        case 2:
            // new-order
            // selected from menu; display new-order
            // input form

            MakeNewOrderForm(TermId, NULL,
                INPUT_FORM, szBuffer);

                break;
        case 3:
            //
            // payment selected from menu; display payment
            // input form

            MakePaymentForm(TermId, NULL,
                INPUT_FORM, szBuffer);

                break;
        case 4:
            //
            // delivery selected from menu; display delivery
            // input form

            MakeDeliveryForm(TermId, NULL,
                INPUT_FORM, szBuffer);

                break;
        case 5:
            // order-
            // status selected from menu; display order-status
            // input form

            MakeOrderStatusForm(TermId,
                NULL, INPUT_FORM, szBuffer);

                break;
        case 6:
            // stock-
            // level selected from menu; display stock-level
            // input form

            MakeStockLevelForm(TermId,
                NULL, INPUT_FORM, szBuffer);

                break;
        case 7:
            //

            ExitCmd

            TermDelete(TermId);

            WelcomeForm(pECB, szBuffer);

                break;
        case 8:

            SubmitCmd(pECB, szBuffer);

                break;
        case 9:
            // menu

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

                break;
        case 10:
            //

            CMD=Clear

                // resets
                // all connections; should only be used when no
                // other connections are active

```

```

        TermDeleteAll();
        TermInit();

        WelcomeForm(pECB, szBuffer);

                break;
        case 11:
            //

            CMD=Stats

            StatsCmd(pECB, szBuffer);

                break;
            }
        }
        catch (CBaseErr *e)
        {
            ErrorForm( pECB, e-
                >ErrorType(), e->ErrorNum(), TermId, iSyncId,
                e->ErrorText(), szBuffer );

                delete e;
        }
        catch (...)
        {
            ErrorForm( pECB,
                ERR_TYPE_WEBDLL, 0, TermId, iSyncId, "Error:
                Unhandled exception in Web Client.", szBuffer );
        }

#ifdef ICECAP
        StopCAP();
#endif

        lpbSize = strlen(szBuffer);
        wsprintf(szHeader1,

            "Content-Type: text/html\r\n"

            "Content-Length: %d\r\n"

            "Connection: Keep-Alive\r\n\r\n",
            lpbSize);

        strcat( szHeader1, szBuffer );

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

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

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

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

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

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

```

```

        0, // event category
        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
    txnDelRec;

    DWORD
    index;

    HANDLE
    handles[2];

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

    assert(txnDelilog != NULL);

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

                //

```

```

        EnterCriticalSection(&hConnectCriticalSection);

        Sleep(Reg.dwConnectDelay);

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

            szMyComputerName,
            Reg.szDbName,

            Reg.szSPPrefix,
            Reg.bCallNoDuplicatesNewOrder );

        LeaveCriticalSection(&hConnectCriticalSection);
    }
    else
    {
        if
            (Reg.eDB_Protocol == DBLIB)

            pTxn = pCTPCC_DBLIB_new(
                Reg.szDbServer, Reg.szDbUser,
                Reg.szDbPassword, szMyComputerName,
                Reg.szDbName );
            }
        pDeliveryData =
            pTxn->BuffAddr_Delivery();
    }
    catch (CBaseErr *e)
    {
        char szTmp[1024];
        wsprintf( szTmp,
            "Error in Delivery Txn thread. Could not connect
            to database. "

            "%s. Server=%s, User=%s,
            Password=%s, Database=%s",

            e->ErrorText(), Reg.szDbServer,
            Reg.szDbUser, Reg.szDbPassword,
            Reg.szDbName );

        WriteMessageToEventLog( szTmp );
        delete e;
        goto ErrorExit;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Un
            handled 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(&DelBuffCritical
            Section);

            delivery =
            *(pDelBuff+dwDelBuffBusyIndex);

            dwDelBuffFreeCount++;

            dwDelBuffBusyIndex++;

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

            dwDelBuffBusyIndex

            = 0;

            LeaveCriticalSection(&DelBuffCritical
            Section);

            pDeliveryData->w_id =
            delivery.w_id;

            pDeliveryData->o_carrier_id =
            delivery.o_carrier_id;

            txnDeliRec.w_id = pDeliveryData-
            >w_id;

            txnDeliRec.o_carrier_id =
            pDeliveryData->o_carrier_id;

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

            GetLocalTime( &trans_start );

            pTxn->Delivery();

            GetLocalTime( &trans_end );

```

```

            //log txn
            txnDeliRec.TxnStatus =
            ERR_SUCCESS;

            for (int i=0; i<10; i++)

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

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

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

            if (txnDeliLog != NULL)

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

                e->ErrorTypeStr(),
                e->ErrorNum(), e->ErrorText() );

                WriteMessageToEventLog( szTmp );

                // log
                the error txn

                txnDeliRec.TxnStatus = e-
                >ErrorType();

                if
                (txnDeliLog != NULL)

                txnDeliLog-
                >WriteToLog(&txnDeliRec);

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

                WriteMessageToEventLog(TEXT("U
                nhandled exception caught in
                DeliveryWorkerThread."));
            }
        }
    }
    ErrorExit:
    if (Reg.dwConnectDelay > 0)
    {
        // Synchronize
        disconnect (for VIA)
        //

        EnterCriticalSection(&hConnectCritical
        Section);

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

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

```

```

}
}
break;
}

/* FUNCTION: void WelcomeForm
*
*/

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

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

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

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

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

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

        "</PRE></font>"

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

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

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

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

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

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

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

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

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

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

```

```

"Max Connections
= <B>%d</B><BR>"

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

"Max Pending
Deliveries = <B>%d</B><BR>"

szTxnMonNames[Reg.eTxnMon],
szDBNames[Reg.eDB_Protocol],

Reg.dwMaxConnections,
dwNumDeliveryThreads, dwDelBuffSize );
strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

"</PRE></font>"

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

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

```

```

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

"</PRE></font>"

, Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, Reg.szDbName );
strcat( szBuffer, szTmp);

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

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

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

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

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

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

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

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

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

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

```

```

if ( strcmp( szVersion,
WEBCLIENT_VERSION ) ) throw new
CWEBCLNT_ERR( ERR_VERSION_MISMATCH );

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

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

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

iNewTerm = TermAdd();

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

try
{
    if (Reg.eTxnMon ==
TUXEDO)

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

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

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

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

```

```

szDatabase, Reg.szSPPrefix,

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

Term.pClientData[iNewTerm].pTxn
= pCTPCC_DBLIB_new( szServer, szUser,
szPassword, szMyComputerName, szDatabase );
}
catch (...)
{

TermDelete(iNewTerm);
throw;
// pass exception upward
}

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

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

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

EnterCriticalSection(&TermCriticalSe
ction);

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

iTotal++;
}

LeaveCriticalSection(&TermCriticalS
ection);

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_RAN
GE,
"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_KE
Y,
"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,

DLL="
"Load of DLL failed.

},
{
ERR_MAX_CONNECTIONS_EXCEED
ED,
"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_INV
ALID,
"New
Order customer id invalid data type, range = 1 to
3000." },
{
ERR_NEWORDER_CUSTOMER_KEY,

"New Order missing Customer key
\"CID*\"."

},
{
ERR_NEWORDER_DISTRICT_INVAL
ID,
"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_WITHOU
T_SUPPW,
"New Order Item_Id
field entered without a corresponding Supp_W."

},
{
ERR_NEWORDER_MISSING_IID_KE
Y,
"New
Order missing Item Id key \"IID*\"."

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

},

```



```

    {
      ERR_NEWORDER_MISSING_SUPPW
      _KEY, "New
      Order missing Supp_W key \"SP#*\"."
    },
    {
      ERR_NEWORDER_NOITEMS_ENTER
      ED, "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_INVALI
      D, "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_L
      ONG, "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_KE
      Y, "Payment missing Amount key
      \"HAM*\"."
    },
    {
      ERR_STOCKLEVEL_MISSING_THRE
      SHOLD_KEY, "Stock Level; missing Threshold key
      \"TT*\"."
    },
    {
      ERR_STOCKLEVEL_THRESHOLD_IN
      VALID, "Stock Level;
      Threshold value must be in the range = 1 - 99."
    },
    {
      ERR_STOCKLEVEL_THRESHOLD_RA
      NGE, "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
* key value to look for
*
* char
* *pValue
* character array into which to place
key's value
*
* int
* iMax
* maximum length of
key value array.
*
* WEBERROR      err
*
* error value to throw

```

```

* RETURNS:      nothing.
* ERROR:       if (the pKey value is
not found) then
*
* if (err == 0)
*
* return
*
* (empty string)
*
* else
*
* throw
*
* CWEBCLNT_ERR(err)
*
* COMMENTS:    http keys are
formatted either KEY=value& or KEY=value\0.
This DLL formats
*
* TPC-C input fields in such a manner
that the keys can be extracted in the
*
* above manner.
*/
void GetKeyValue(char **pQueryString, char
*pKey, char *pValue, int iMax, WEBERROR err)
{
    char *ptr;
    if ( !(ptr=strstr(*pQueryString,
pKey)) )
        goto ErrorExit;
    ptr += strlen(pKey);
    if ( *ptr != '=' )
        goto ErrorExit;
    ptr++;
    iMax--; // one position is for
terminating null
    while( *ptr && *ptr != '&' && iMax)
    {
        *pValue++ = *ptr++;
        iMax--;
    }
    *pValue = 0; // terminating null
    *pQueryString = ptr;
    return;
ErrorExit:
    if (err != NO_ERR)
        throw new
        CWEBCLNT_ERR( err );
    *pValue = 0; // return empty
result string
}

/* FUNCTION: GetIntKeyValue
*
* PURPOSE: This function parses a http
formatted string for a specific key value.
*
* ARGUMENTS:    char
                  *pQueryString
                  http string from client browser
*
* char
* *pKey
* key value to look for
*
* WEBERROR      NoKeyErr      error
* value to throw if key not found

```

```

*
* WEBERROR
* NotIntErr      error
* value to throw if value not numeric
* RETURNS:      integer
*
* ERROR:       if (the pKey value is
not found) then
*
* if (NoKeyErr !=
NO_ERR)
*
* throw
*
* CWEBCLNT_ERR(err)
*
* else
*
* return 0
*
* else if (non-numeric char found)
then
*
* if (NotIntErr !=
NO_ERR) then
*
* throw
*
* CWEBCLNT_ERR(err)
*
* else
*
* return 0
*
* COMMENTS:    http keys are
formatted either KEY=value& or KEY=value\0.
This DLL formats
*
* TPC-C input fields in such a manner
that the keys can be extracted in the
*
* above manner.
*/
int GetIntKeyValue(char **pQueryString, char
*pKey, WEBERROR NoKeyErr, WEBERROR
NotIntErr)
{
    char *ptr0;
    char *ptr;
    if ( !(ptr=strstr(*pQueryString,
pKey)) )
        goto ErrorNoKey;
    ptr += strlen(pKey);
    if ( *ptr != '=' )
        goto ErrorNoKey;
    ptr++;
    ptr0 = ptr; //
remember starting point
    // scan string until a terminator
(null or &) or a non-digit
    while( *ptr && *ptr != '&' &&
isdigit(*ptr) )
        ptr++;
    // make sure we stopped scanning
for the right reason
    if ((ptr0 == ptr) || (*ptr && *ptr !=
'&'))
    {
        if (NotIntErr !=
NO_ERR)
            throw
            new CWEBCLNT_ERR( NoKeyErr );
        return 0;
    }
}

```

```

    }

    *pQueryString = ptr;
    return atoi(ptr0);

ErrorNoKey:
    if (NoKeyErr != NO_ERR)
        throw new
CWEBCLNTErr( 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(&TermCriticalSe
ction);

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

    Term.pClientData = NULL;
    Term.pClientData =
(PCLIENTDATA)malloc(Term.iNumEntries *
sizeof(CLIENTDATA));
    if (Term.pClientData == NULL)
    {
        LeaveCriticalSection(&TermCriticalS
ection);
        throw new
CWEBCLNTErr( 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(&TermCriticalS
ection);
}

/* 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(&TermCriticalSe
ction);

    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(&TermCriticalS
ection);
    }

/* 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(&TermCriticalSe
ction);
    if (Term.iFreeList != 0)
    {
        // position is available
        iNewTerm =
Term.iFreeList;
        Term.iFreeList =
Term.pClientData[iNewTerm].iNextFree;

        Term.pClientData[iNewTerm].iNext
Free = -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(&TermCriticalS
ection);
            throw
new CWEBCLNTErr(
ERR_MAX_CONNECTIONS_EXCEEDED );
        }

        Term.pClientData[iNewTerm].iTickC
ount = GetTickCount();
        Term.pClientData[iNewTerm].iSynC
id = Term.iMasterSyncId++;
        Term.pClientData[iNewTerm].pTxn
= NULL;

        LeaveCriticalSection(&TermCriticalS
ection);
        return iNewTerm;
    }

/* FUNCTION: TermDelete
 *
 * PURPOSE: This function makes a terminal id
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(&TermCriticalSe
ction);

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

        LeaveCriticalSection(&TermCriticalS
ection);
    }
}

/* FUNCTION: MakeErrorForm
 */

```

```

void ErrorForm(EXTENSION_CONTROL_BLOCK
*pECB, int iType, int iErrorNum, int iTermId, int
iSyncId, char *szErrorText, char *szBuffer )
{
    wsprintf(szBuffer,
        "<HTML><HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
        "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"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)
{
    wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C
Main Menu</TITLE></HEAD><BODY>"
        "Select Desired
Transaction.<BR><HR>"
        "<FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"TERMINID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"%d\">"
        "<PRE><font
face=\"Courier\">
Level<BR>"
    );
}

```

```

        "<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 = wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C
Stock Level</TITLE></HEAD><FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
        "<INPUT
TYPE=\"hidden\" NAME=\"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>"
                "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                "<INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"
                "</FORM></HTML>" );
        }
        else
        {
            wsprintf(szForm+c,
                "Stock
Level Threshold: %2.2d<BR> <BR>"
                "low
stock: %3.3d</font> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR>"
                "<BR>"
                "<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=\"..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>";

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

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

    c = sprintf(szForm,

" <HTML><HEAD><TITLE>TPC-C
New Order</TITLE></HEAD><BODY>"
" <FORM
ACTION=\"tpcc.dll\" METHOD=\"GET\">"
" <INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"
" <INPUT
TYPE=\"hidden\" NAME=\"ERROR\"
VALUE=\"0\">"
" <INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"
" <INPUT
TYPE=\"hidden\" NAME=\"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 +=
sprintf(szForm+c, "Warehouse: %6.6d ",
Term.pClientData[iTermId].w_id );

        strcpy( szForm+c,
"District:
<INPUT NAME=\"DID*\" SIZE=1>
Date:<BR>"

"Customer: <INPUT
NAME=\"CID*\" SIZE=4> Name:
Credit: %Disc:<BR>"

"Order
Number: Number of Lines: W_tax:
D_tax:<BR> <BR>"

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

"
<INPUT NAME=\"SP00*\" SIZE=4> <INPUT
NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"

```

```

"
<INPUT NAME=\"SP01*\" SIZE=4> <INPUT
NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
<INPUT NAME=\"SP02*\" SIZE=4> <INPUT
NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP03*\" SIZE=4> <INPUT
NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP04*\" SIZE=4> <INPUT
NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP05*\" SIZE=4> <INPUT
NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP06*\" SIZE=4> <INPUT
NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP07*\" SIZE=4> <INPUT
NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP08*\" SIZE=4> <INPUT
NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP09*\" SIZE=4> <INPUT
NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP10*\" SIZE=4> <INPUT
NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP11*\" SIZE=4> <INPUT
NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP12*\" SIZE=4> <INPUT
NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP13*\" SIZE=4> <INPUT
NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
"
<INPUT NAME=\"SP14*\" SIZE=4> <INPUT
NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"

"Execution Status:
Total:<BR>"

" </font></PRE><HR>"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
" <INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Menu\">"

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

    }
    else
    {
        c +=
sprintf(szForm+c, "Warehouse: %6.6d
District: %2.2d Date: ",

```

```

pNewOrderData->w_id,
pNewOrderData->d_id);

    if ( bValid )
    {
        c +=
sprintf(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>",
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_name,
pNewOrderData->ol_quantity,
pNewOrderData->ol_stock,
pNewOrderData->ol_brand_generic,
pNewOrderData->ol_i_price,
pNewOrderData->ol_amount );
    }
    else
    {
        c +=
wprintf(szForm+c,
        "%Disc: <BR>"
        "Order Number: %8.8d Number of
        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 +=
wprintf(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..\">"

```

```

) );
}
}
/* 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 = wprintf(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 +=
wprintf(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 +=
wprintf(szForm+c,
" <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 +=
wprintf(szForm+c,
" <BR>
<BR>Warehouse: %6.6d
District: %2.2d<BR>"
"%-20s
%-20s<BR>"
"%-20s
%-20s<BR>"
"%-20s
%-2s %5.5s-%4.4s %-20s %-2s %5.5s-
%4.4s<BR> <BR>"
"Customer: %4.4d Cust-
Warehouse: %6.6d Cust-District: %2.2d<BR>"
"Name:
%-16s %-2s %-16s Since: %2.2d-%2.2d-
%4.4d<BR>"
"
%-20s Credit: %-2s<BR>"
'
Term.pClientData[iTermId].w_id, pPaymentData-
>d_id
pPaymentData->w_street_1, pPaymentData-
>d_street_1
pPaymentData->w_street_2, pPaymentData-
>d_street_2
pPaymentData->w_city, pPaymentData-
>w_state, pPaymentData->w_zip,
pPaymentData->w_zip+5
pPaymentData->d_city, pPaymentData->d_state,
pPaymentData->d_zip, pPaymentData->d_zip+5

```

```
pPaymentData->c_id, pPaymentData->c_w_id,
    pPaymentData->c_d_id
    pPaymentData->c_first, pPaymentData-
->c_middle, pPaymentData->c_last
    pPaymentData->c_since.day, pPaymentData-
->c_since.month, pPaymentData-
->c_since.year
    pPaymentData->c_street_1, pPaymentData-
->c_credit
    );
    c += sprintf(szForm+c,
        "
    %-20s           %%Disc: %5.2f<BR>",
        pPaymentData->c_street_2,
    100.0*pPaymentData->c_discount);
    c +=
    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 );
    else
        strcpy(szForm+c, "Cust-Data:
    <BR> <BR> <BR> <BR>");
        strcat(szForm,
            " <BR></font></PRE><HR>"
        " <INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..NewOrder..\">"
    )
    )
```

```
<INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..Payment..\">
    <INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..Delivery..\">
    <INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..Order-Status..\">
    <INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..Stock-Level..\">
    <INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"..Exit..\">
    </BODY></FORM></HTML>";
    }
/* FUNCTION: MakeOrderStatusForm
 *
 * COMMENTS: The internal client
    buffer is created when the terminal id is assigned
    and should not
 *
    be freed except when the client
    terminal id is no longer needed.
 */
void MakeOrderStatusForm(int iTermId,
    ORDER_STATUS_DATA *pOrderStatusData,
    BOOL bInput, char *szForm)
{
    int i, c;
    static char szBR[] = " <BR> <BR>
    <BR> <BR> <BR> <BR> <BR> <BR> <BR>
    <BR> <BR> <BR> <BR> <BR> <BR>";
    c = 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=\"TERMINID\"
    VALUE=\"\"%d\">"
        "<INPUT
    TYPE=\"hidden\" NAME=\"SYNCID\"
    VALUE=\"\"%d\">"
        "<PRE><font
    face=\"Courier\">
        Order-
    Status<BR>"
        "Warehouse: %6.6d
    ",
        ORDER_STATUS_FORM, iTermId,
    Term.pClientData[iTermId].iSyncId,
    Term.pClientData[iTermId].w_id);
```

```
    if ( bInput )
    {
        strcpy(szForm+c,
            "District:
    <INPUT NAME=\"DID*\" SIZE=1><BR>"
            "Customer: <INPUT
    NAME=\"CID*\" SIZE=4> Name:
    <INPUT NAME=\"CLT*\" SIZE=23><BR>"
            "Cust-
    Balance:<BR> <BR>"
            "Order-
    Number:
    Entry-Date:
    Carrier-Number:<BR>"
            "Supply-
    W Item-Id Qty Amount Delivery-
    Date<BR> <BR> <BR> <BR> <BR>
    " <BR>
    <BR> <BR> <BR> <BR> <BR> <BR>
    <BR> <BR> <BR></font></PRE>"
            "<HR><INPUT TYPE=\"submit\"
    NAME=\"CMD\" VALUE=\"Process\"><INPUT
    TYPE=\"submit\" NAME=\"CMD\"
    VALUE=\"Menu\">"
            "</BODY></FORM></HTML> " );
        else
        {
            c +=
            sprintf(szForm+c,
                "District:
                %2.2d<BR>"
                "Customer: %4.4d Name: %6.6s
                %2s %16s<BR>",
                pOrderStatusData->d_id,
                pOrderStatusData->c_id,
                pOrderStatusData->c_first,
                pOrderStatusData->c_middle, pOrderStatusData-
->c_last);
            c +=
            sprintf(szForm+c, "Cust-Balance: %9.2f<BR>
            <BR>",
                pOrderStatusData->c_balance);
            c +=
            sprintf(szForm+c, "Order-
            Number: %8.8d Entry-Date: %2.2d-%2.2d-
            %4.4d %2.2d:%2.2d:%2.2d Carrier-Number:
            %2.2d<BR>"
                "Supply-
            W Item-Id Qty Amount Delivery-
            Date<BR>",
                pOrderStatusData->o_id,
                pOrderStatusData->o_entry_d.day,
                pOrderStatusData-
->o_entry_d.month,
                pOrderStatusData->o_entry_d.year,
                pOrderStatusData-
->o_entry_d.hour,
```

```

    pOrderStatusData-
>o_entry_d.minute,

    pOrderStatusData-
>o_entry_d.second,

    pOrderStatusData->o_carrier_id);

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

    pOrderStatusData-
>OL[i].ol_supply_w_id,

    pOrderStatusData->OL[i].ol_i_id,

    pOrderStatusData-
>OL[i].ol_quantity,

    pOrderStatusData-
>OL[i].ol_amount,

    pOrderStatusData-
>OL[i].ol_delivery_d_day,

    pOrderStatusData-
>OL[i].ol_delivery_d_month,

    pOrderStatusData-
>OL[i].ol_delivery_d_year);
    }

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

    strcpy(szForm+c,

    "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"

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

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

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

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

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

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

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

```

```

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

    c = wsprintf(szForm,

    "<HTML><HEAD><TITLE>TPC-C
Delivery</TITLE></HEAD><BODY>"

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

    "<INPUT
TYPE=\"hidden\" NAME=\"STATUSID\"
VALUE=\"%d\">"

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

    "<INPUT
TYPE=\"hidden\" NAME=\"FORMID\"
VALUE=\"%d\">"

    "<INPUT
TYPE=\"hidden\" NAME=\"TERMINID\"
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_B
LOCK *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_B
OCK *pECB, int iTermId, char *szBuffer)
{
    PPAYMENT_DATA
    pPayment;

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

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

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

/* FUNCTION: ProcessOrderStatusForm
*
* PURPOSE: This function gets and validates the
input data from the Order Status
*
    form
filling in the required input variables. It then calls
the
*
    SQLOrderStatus transaction,
constructs the output form and writes it
*
    back to
client browser.
*
* ARGUMENTS:
    EXTENSION_CONTROL_BLOCK
    *pECB    passed in structure
pointer from inetsrv.
*
    int
    iTermId    client browser
terminal id
*
*/
void
ProcessOrderStatusForm(EXTENSION_CONTROL_
BLOCK *pECB, int iTermId, char *szBuffer)
{
    PORDER_STATUS_DATA
    pOrderStatus;

```

```

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

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

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

/* FUNCTION: ProcessDeliveryForm
*
* PURPOSE: This function gets and validates the
input data from the delivery form
*
    filling in
the required input variables. It then calls the
PostDeliveryInfo
*
    Api, The
client is then informed that the transaction has
been posted.
*
* ARGUMENTS:
    EXTENSION_CONTROL_BLOCK
    *pECB    passed in structure
pointer from inetsrv.
*
    int
    iTermId    client browser
terminal id
*
*/
void
ProcessDeliveryForm(EXTENSION_CONTROL_B
LOCK *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_B
LOCK *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         items;
    short       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++;
            }
            //
            nothing entered for supply warehouse, so item id
            and qty must also be blank

            GetKeyValue(&ptr, szIID[i], szTmp,
sizeof(szTmp),
ERR_NEWORDER_MISSING_IID_KEY);
            if (
szTmp[0] )

            throw new CWEBCLNT_ERR(
ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

            GetKeyValue(&ptr, szQty[i], szTmp,
sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
            if (
szTmp[0] )

            throw new CWEBCLNT_ERR(
ERR_NEWORDER_QTY_WITHOUT_SUPPW );
        }
        if ( items == 0 )
            throw new
CWEBCLNT_ERR(
ERR_NEWORDER_NOITEMS_ENTERED );

```

```

        }
        pNewOrderData->o_ol_cnt = items;

/* FUNCTION: GetPaymentData
*
* PURPOSE: This function extracts and validates
the payment form data from an http command
string.
*
* ARGUMENTS:          LPSTR
                    lpszQueryString
                    client browser http
command string
*
                    PAYMENT_DATA
                    *pPaymentData
                    pointer to payment data structure
*/

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

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

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

        throw
new CWEBCLNT_ERR(
ERR_PAYMENT_CUSTOMER_INVALID );
        pPaymentData->c_id
= atoi(szTmp);
    }

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

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

```

```

        _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_jd =
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->
GetKeyValue(&ptr,
"CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_MISSING_CID_CLT );

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

        strcpy(pOrderStatusData->c_last,
szTmp);
        // pad with spaces so
that the client layer doesn't have to do it
        // before passing
parameters to stored procedure
        iLen =
strlen(pOrderStatusData->c_last);

        memset(pOrderStatusData->c_last
+ iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData->
>c_last[LAST_NAME_LEN] = 0;
        }
        else
        {
            // parse customer id
and verify that last name was NOT entered
            if ( !IsNumeric(szTmp) )
                throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_INVALID );
        pOrderStatusData->
>c_id = atoi(szTmp);
        GetKeyValue(&ptr,
"CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw
new CWEBCLNT_ERR(
ERR_ORDERSTATUS_CID_AND_CLT );
        }
    }

/* FUNCTION: BOOL IsNumeric(char *ptr)
*
* PURPOSE: This function determines if a string
is numeric. It fails if any characters other
than
numeric and null terminator are present.
*
* ARGUMENTS:    char        pointer
                *ptr
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        pointer
                *ptr
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
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 102
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

common/src/ReadRegistry.h

```

/* FILE:
   ReadRegistry.h

   Microsoft TPC-C Kit Ver. 4.20.000

   Copyright Microsoft, 1999
   All Rights Reserved

   not audited

   PURPOSE: Header for registry
   related code.

   Change history:
   * 4.20.000 - first
   version
   */

enum DBPROTOCOL { Unspecified, ODBC, DBLIB
};
const char *szDBNames[] = { "Unspecified",
"ODBC", "DBLIB" };

enum TXNMON { None, TUXEDO, ENCINA, COM
};
const char *szTxnMonNames[] = { "NONE",
"TUXEDO", "ENCINA", "COM" };

//This structure defines the data necessary to
keep distinct for each terminal or client
connection.
typedef struct _TPCCREGISTRYDATA
{
    enum DBPROTOCOL eDB_Protocol;
    enum TXNMON eTxnMon;
    BOOL bCOM_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    DWORD
dwNumberOfDeliveryThreads;
    char szPath[128];
    char szDbServer[32];
    char szDbName[32];
    char szDbUser[32];
    char szDbPassword[32];
    wchar_t szSPPrefix[32];
    //tpcc_odbc.dll stored procedures
prefix
    DWORD dwConnectDelay; // delay
in ms to use in pacing connection open and close

```

```

    BOOL bCallNoDuplicatesNewOrder;
    // whether to check for non-
duplicate item ids and call a different New Order
function
TPCCREGISTRYDATA *pTPCCREGISTRYDATA;

    BOOL ReadTPCCRegistrySettings(
TPCCREGISTRYDATA *pReg );
);

```

ReadRegistry.cpp

```

/* FILE:
   READREGISTRY.CPP

   Microsoft TPC-C Kit Ver. 4.20.000

   Copyright Microsoft, 1999
   All Rights Reserved

   not yet audited

   PURPOSE: Implementation for
TPC-C Tuxedo class.
   Contact: Charles Levine
(clevine@microsoft.com)

   Change history:
   * 4.20.000 - first
   version
   */

/* FUNCTION: ReadTPCCRegistrySettings
   * PURPOSE: This function reads the NT registry
   for startup parameters. There parameters are
   under
the TPCC key.
   * RETURNS FALSE = no errors
   TRUE =
error reading registry
   */
BOOL ReadTPCCRegistrySettings(
TPCCREGISTRYDATA *pReg )
{
    HKEY hKey;
    DWORD size;
    DWORD type;
    DWORD dwTmp;
    char szTmp[256];

    if (
RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, KEY_READ,
&hKey) != ERROR_SUCCESS )
        return TRUE;

    // determine database protocol to
use; may be either ODBC or DBLIB
    pReg->eDB_Protocol =
Unspecified;
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey,
"DB_Protocol", 0, &type, (BYTE *)&szTmp, &size)
== ERROR_SUCCESS )
    {

```

```

        if ( !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

//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
#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_BCONN

17
//Benchcraft
connection class
#define ERR_TYPE_TPCC_CONN

18
//Benchcraft connection class

```

```

#define ERR_TYPE_ENCINA
//Encina error
19
#define ERR_TYPE_COMPONENT
//error from COM component
20
#define ERR_TYPE_RTE
//Benchcraft rte
21
#define ERR_TYPE_AUTOMATION
//Benchcraft automation errors
22
#define ERR_TYPE_DRIVER
//Driver engine errors
23
#define ERR_TYPE_RTE_BASE
//Framework errors
24
#define ERR_BUF_OVERFLOW
//Buffer overflow during receive
25
#define ERR_TYPE_SOAP_HTTP
//HTTP/SOAP dll generated error
26
#define ERR_TYPE_OLEDB
//OLE-DB generated error
27
#define ERR_TYPE_TPCC_OLEDB
//error from tpcc ole-db txn module
// TPC-W error types
28
#define ERR_TYPE_TPCW_CONN
//Benchcraft connection class
50
#define ERR_TYPE_TPCW_HTML
//error from TpcwHtml dll
51
#define ERR_TYPE_TPCW_USER
//error from TPC-W user class
52
#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

```

```

#define ERR_TYPE_TPCE_CONN
//TPC-E error types
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(GetModuleHan
dle(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(GetModuleHan
dle(NULL), m_szApp, m_szApp_size);
}
virtual ~CBaseErr(void)
{
if (m_szApp)
delete []
m_szApp;
if (m_szLoc)
delete []
m_szLoc;
};
virtual void Draw(HWND hwnd,
LPCTSTR szStr = NULL)
{
int
j = 0;
char
szTmp[512];
if (szStr)
j =
wsprintf(szTmp, "%s\n", szStr);
if (ErrorNum() !=
INV_ERROR_CODE)
j +=
wsprintf(szTmp+j, "Error = %d\n", ErrorNum());
if (m_szLoc)
j +=
wsprintf(szTmp+j, "Location = %s\n",
GetLocation());
j +=
wsprintf(szTmp+j, "%s\n", ErrorText());
MessageBox(hwnd,
szTmp, m_szApp, MB_OK);
}
char *GetApp(void) { return
m_szApp; }
char *GetLocation(void) { return
m_szLoc; }
virtual int ErrorNum() { return
m_idMsg; }

virtual int ErrorType() = 0; // a
value which distinguishes the kind of error that
occurred

```

```

        virtual char *ErrorTypeStr() = 0; //
text representation of the error type
        virtual char *ErrorText() = 0; // a
string (i.e., human readable) representation of
the error
        virtual int ErrorAction() { return
eNone; } // the function call that caused the
error

protected:
        char          *m_szApp;
        char          *m_szLoc; // code
location where the error occurred
        int
        m_idMsg;

        //short      m_errType;
};

class CSocketErr : public CBaseErr
{
public:
        enum Action
        {
                eNone = 0,
                eSend,
                eSocket,
                eBind,
                eConnect,
                eListen,
                eHost,
                eRecv,
                eGetHostByName,
                eWSACreateEvent,
                eWSASend,

                eWSAGetOverlappedResult,
                eWSARecv,

                eWSAWaitForMultipleEvents,
                eWSAStartup,
                eWSAResetEvent,

                eWSAEnumNetworkEvents,
                eWSAEventSelect,
                eSelect,
                eAccept,
                eNonRetryable
        };

        CSocketErr(Action eAction,
LPCTSTR szLocation = NULL);

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

                Action      m_eAction;
                char        *m_szErrorText;

                int
                ErrorType() { return
ERR_TYPE_SOCKET;};
                char*      ErrorTypeStr() {
return "SOCKET"; }
                char*      ErrorText(void);
                int
                ErrorAction() { return
(int)m_eAction; }

```

```

};

class CSystemErr : public CBaseErr
{
public:
        enum Action
        {
                eNone = 0,
                eTransactNamedPipe,
                eWaitNamedPipe,

                eSetNamedPipeHandleState,
                eCreateFile,
                eCreateProcess,
                eCallNamedPipe,
                eCreateEvent,
                eCreateThread,
                eVirtualAlloc,
                eReadFile = 10,
                eWriteFile,
                eMapViewOfFile,
                eCreateFileMapping,

                eInitializeSecurityDescriptor,

                eSetSecurityDescriptorDacl,
                eCreateNamedPipe,
                eConnectNamedPipe,
                eWaitForSingleObject,
                eRegOpenKeyEx,
                eRegQueryValueEx =
20,

                ebeginthread,
                eRegEnumValue,
                eRegSetValueEx,
                eRegCreateKeyEx,

                eWaitForMultipleObjects,
                eRegisterClassEx,
                eCreateWindow,
                eCreateSemaphore,
                eReleaseSemaphore,
                eFSeek,
                eFRead,
                eFWrite,
                eTmpFile,
                eSetFilePointer,
                eNew,
                eCloseHandle,
                eGetOverlappedResult
        };

        CSystemErr(Action eAction,
LPCTSTR szLocation);

        CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
                int
                ErrorType() { return
ERR_TYPE_OS;};
                char*      ErrorTypeStr() {
return "SYSTEM"; }
                char        *ErrorText(void);
                int
                ErrorAction() { return
(int)m_eAction; }
                void      Draw(HWND hwnd,
LPCTSTR szStr = NULL);

                Action      m_eAction;

private:
                char
                m_szMsg[ERR_MSG_BUF_SIZE];

```

```

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

                int
                ErrorType() {return
ERR_TYPE_MEMORY;};
                char*      ErrorTypeStr() {
return "OUT OF MEMORY"; }
                char*      ErrorText() {return
ERR_INS_MEMORY;};
};

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

                int
                ErrorType() {return
ERR_BUF_OVERFLOW;};
                char*      ErrorTypeStr() {
return "BUFFER OVERFLOW"; }
                char*      ErrorText() {return
ERR_INS_BUF_OVERFLOW;};
};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{
public:
        enum Action
        {
                LoadProfile = 1,

                LoadSchema,

                ValidateProfile,

                SaveProfile,

                LoadFromXML,

                SaveToXML,

                ApplyProcessingInstruction,

                ApplyAttribute,

                ApplyNode
        };

        CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation)
        {
                m_eAction = eAction;
                m_eCode = eCode;
                m_bOverload = true;
                };

                CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation, char *
szMsg)
                {
                m_eAction = eAction;
                m_eCode = eCode;

```

```

strcpy(m_szMsg, szMsg);

m_bOverload = false;
};

virtual int
ErrorType() { return
ERR_TYPE_XML_PROFILE;};
virtual char
*ErrorTypeStr() { return "XML
PROFILE"; };
virtual char
*ErrorText();
virtual int
ErrorCode() { return m_eCode; };
int

ErrorAction() { return
(int)m_eAction; }
//virtual void
Draw(HWND hwnd, LPCTSTR szStr
= NULL)
//{
//
::MessageBox(hwnd, szStr,
m_szLoc, MB_OK);
//};
private:
char
m_szMsg[ERR_MSG_BUF_SIZE];
LPCTSTR m_szLoc;
int
m_eCode;
bool
m_bOverload;
Action
m_eAction;
};

```

common\src\trans.h

```

/* FILE:
ERROR.H
*
* Microsoft TPC-C Kit Ver. 4.20.000
*
* Copyright Microsoft, 1999
* All Rights Reserved
*
* Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
*
* PURPOSE: Header file for error
exception classes.
*
* Change history:
* 4.20.000 - updated
rev number to match kit
* 4.21.000 - fixed bug:
~CBaseErr needed to be declared virtual
*/
#pragma once
#ifdef _INC_STRING

```

```

#endif #include <string.h>

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
error
#define ERR_TYPE_COM 9
//error from COM call
#define ERR_TYPE_TUXEDO 10
//tuxedo error
#define ERR_TYPE_OS 11
//operating system
error
#define ERR_TYPE_MEMORY 12
//memory allocation
error
#define ERR_TYPE_TPCC_ODBC 13
//error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB 14
//error from tpcc dblib txn module
#define ERR_TYPE_DELISRV 15
//delivery server error
#define ERR_TYPE_TXNLOG 16
//txn log error
#define ERR_TYPE_BCCONN 17
//Benchcraft
connection class
#define ERR_TYPE_TPCC_CONN 18
//Benchcraft connection class
#define ERR_TYPE_ENCINA 19
//Encina error
#define ERR_TYPE_COMPONENT 20
//error from COM component
#define ERR_TYPE_RTE 21
//Benchcraft rte
#define ERR_TYPE_AUTOMATION 22
//Benchcraft automation errors
#define ERR_TYPE_DRIVER 23
//Driver engine errors
#define ERR_TYPE_RTE_BASE 24
//Framework errors
#define ERR_BUF_OVERFLOW 25
//Buffer overflow during receive

```



```

#define ERR_TYPE_SOAP_HTTP
                26
                //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
                27
                //OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB
                28
                //error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
                50
                //Benchmark 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"
#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
        = GetLastErr(); //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(GetModuleHan
dle(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(GetModuleHan
dle(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,
eWaitForSingleObject,
eRegOpenKeyEx,
eRegQueryValueEx =
20,

ebeginthread,
eRegEnumValue,
eRegSetValueEx,
eRegCreateKeyEx,

eWaitForMultipleObjects,
eRegisterClassEx,
eCreateWindow,
eCreateSemaphore,
eReleaseSemaphore,
eFSeek,
eFRead,
eFWrite,
eTmpFile,
eSetFilePointer,
eNew,
eCloseHandle,
eGetOverlappedResult
};

CSystemErr(Action eAction,
LPCTSTR szLocation);

CSystemErr(int iError, Action
eAction, LPCTSTR szLocation);
int
ErrorType() { return
ERR_TYPE_OS;};
char* ErrorTypeStr() {
return "SYSTEM"; }
char *ErrorText(void);
int
ErrorAction() { return
(int)m_eAction; }
void Draw(HWND hwnd,
LPCTSTR szStr = NULL);

Action m_eAction;

private:
    char
m_szMsg[ERR_MSG_BUF_SIZE];
};

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

    int
    ErrorType() {return
ERR_TYPE_MEMORY;};
    char* ErrorTypeStr() {
return "OUT OF MEMORY"; }
    char* ErrorText() {return
ERR_INS_MEMORY;};
};

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

    int
    ErrorType() {return
ERR_BUF_OVERFLOW;};
    char* ErrorTypeStr() {
return "BUFFER OVERFLOW"; }
};

```

```

char* ErrorText() {return
ERR_INS_BUF_OVERFLOW;};

// Exception type for XML profiles
class CXMLProfileErr : public CBaseErr
{
public:
    enum Action
    {
        LoadProfile = 1,
        LoadSchema,
        ValidateProfile,
        SaveProfile,
        LoadFromXML,
        SaveToXML,
        ApplyProcessingInstruction,
        ApplyAttribute,
        ApplyNode
    };

    CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation)
    {
        m_eAction = eAction;
        m_eCode = eCode;
        m_bOverload = true;
    };

    CXMLProfileErr(Action
eAction, int eCode, LPCTSTR szLocation, char *
szMsg)
    {
        m_eAction = eAction;
        m_eCode = eCode;
        strcpy(m_szMsg, szMsg);
        m_bOverload = false;
    };

    virtual int
    ErrorType() { return
ERR_TYPE_XML_PROFILE;};
    virtual char
    *ErrorTypeStr() { return "XML
PROFILE"; };
    virtual char
    *ErrorText();
    virtual int
    ErrorCode() { return m_eCode; };
    int

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

```

```

private:
    char
    m_szMsg[ERR_MSG_BUF_SIZE];
    LPCTSTR m_szLoc;
    int
    m_eCode;
    bool
    m_bOverload;
    Action
    m_eAction;
};

```

Common\src\txn_base.h

```

/* FILE:
   TXN_BASE.H
*
* Microsoft TPC-C Kit Ver. 4.20.000
*
* Copyright Microsoft, 1999
* All Rights Reserved
*
* Version 4.10.000 audited by
* Richard Gimarc, Performance Metrics, 3/17/99
*
* PURPOSE: Header file for TPC-C
* txn class implementation.
*
* Change history:
* 4.20.000 - updated
* rev number to match kit
*/

#pragma once

// need to declare functions for import, unless
// define has already been created
// by the DLL's .cpp module for export.
#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
PORORDER_STATUS_DATA
    BuffAddr_OrderStatus()
    = 0;

```

```

virtual void NewOrder
    () = 0;
virtual void Payment
    () = 0;
virtual void Delivery
    () = 0;
virtual void StockLevel
    () = 0;
virtual void
OrderStatus () = 0;
};

```

\common\txnlog\include\rtetime.h

```

/* FILE: rtetime.h : header file
* Copyright 1997 Microsoft Corp., All rights
* reserved.
*
* Source code licensed to Tandem Computers
* for Internal
* use only. Redistribution of source or object
* files or
* any derivative works is prohibited. By
* agreement, this
* notice may not be removed.
*
* Authors: Charles Levine, Philip Durr
*
* Microsoft Corp.
*/

//FILE: RTETIME.H

#define MAX_JULIAN_TIME
    0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME __int64
#define TC_TIME DWORD
extern "C"
{
    BOOL
    InitJulianTime(LPSYSTEMTIME
    lpInitTime);
    JULIAN_TIME
    GetJulianTime(void);
    DWORD
    MyTickCount(void);
    void
    GetJulianAndTC(JULIAN_TIME
    *pJulian, DWORD *pTC);
    JULIAN_TIME
    ConvertTo64BitTime(int iYear, int
    iMonth, int iDay, int iHour, int iMinute, int
    iSecond);
    JULIAN_TIME
    Get64BitTime(LPSYSTEMTIME
    lpInitTime);
    int
    JulianDay( int yr, int mm, int dd );
    void
    JulianToTime(JULIAN_TIME
    julianTS, int* yr, int* mm, int* dd, int *hh, int
    *mi, int *ss );
    void
    JulianToCalendar( int
    day, int* yr, int* mm, int* dd );
}

```

\common\txnlog\include\spinlock.h

```

/* FILE: SPINLOCK.H
*
* Copyright 1997 Microsoft Corp., All rights
* reserved.
*
* Source code licensed to Tandem Computers
* for Internal
* use only. Redistribution of source or object
* files or
* any derivative works is prohibited. By
* agreement, this
* notice may not be removed.
*
* Authors: Mike Parkes, Charles Levine, Philip
* Durr
*
* 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
    * held for a very short time.
    When a lock is taken a memory
    * location is exchanged. All other
    threads that want this
    * lock wait by spinning and
    sometimes sleeping on a semaphore
    * until it becomes free again.
    The only other choice is not
    * to wait at all and move on to do
    something else. This
    * module should normally be
    used in conjunction with cache
    * aligned memory in minimize
    cache line misses.
    *****/

    class Spinlock
    {
    // Private data.
    HANDLE
    Semaphore;
    volatile LONG
    m_Spinlock;
    volatile LONG
    Waiting;

#ifdef _DEBUG
    //
    Counters for debugging builds.
    volatile
    LONG TotalLocks;
    volatile
    LONG TotalSleeps;
#endif
}

```

```

volatile
LONG TotalSpins;
volatile
LONG TotalWaits;
#ifdef
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
wake up 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
of txn TxnStartT0; // start
BYTE
TxnType;
// = TXN_REC_TYPE_CONTROL
BYTE
TxnSubType;
// depends on TxnType
// end of common
header
DWORD Len;
// number of bytes after this field
} TXN_RECORD_CONTROL,
*PTXN_RECORD_CONTROL;
// TPC-C Txn Record Layout:
//
//TxnStartT0 is a Julian timestamp
corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of
response time. Deltas
//are in milliseconds. Note that if RTDelay >
0, then the txn was
//delayed by this amount. The delay occurs at
the beginning of the
//response time. So if RTDelay > 0, then the
txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
// time -->
// |--- Menu ---|--- Keying ---|--- Response ---
|--- Think ---|
// <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4
-> <- DeltaT3 ->
// ^
// ^ TxnStartT0
//
//RTDelay is the amount of response time
delay included in DeltaT4.
//RTDelay is recorded per txn because this
value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code. It is
used to indicate errors.
//For example, in the New Order txn, 1% of
txns abort. TxnStatus will
//reflect this.
typedef struct
_TXN_RECORD_TPCC
{
// common header;
must exactly match TXN_RECORD_HEADER
JULIAN_TIME
TxnStartT0; // start
of txn
BYTE
TxnType;
// = TXN_REC_TYPE_TPCC
BYTE
TxnSubType;
// depends on TxnType
// end of common
header

```

```

int DeltaT1;
// think 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
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
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?
*PTXN_RECORD_TPCW_USER;

#define USER_INDEX_NBITS
22
#define USER_INDEX_MASK
0x003ffff
//lower 22 bits mask for user field
in TPCW record
#define USER_SESSION_MASK
0xffc00000 //upper
10 bits mask for user field in TPCW record
#define USER_CREATE_REC
254
//subtype for the control record
written when a user is created

#define TXN_LOG_VERSION
2
#define TXN_DATA_START
4096 // offset
in log file where log records start
#define
TXN_LOG_EYE_CATCHER "BC"
// signature bytes at the start of log
file

////////////////////////////////////
////////////////////////////////////
// The transaction log has a header
as the first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
        char
        EyeCatcher[2];
// signature bytes; should always
be "BC"
        int
        LogVersion; // set to
TXN_LOG_VERSION
        JULIAN_TIME
        BeginTxnTS;
// timestamp of first (lowest) txn
start
        JULIAN_TIME
        EndTxnTS;
// timestamp of last (highest) txn
completion time
        int
        iRecCount; //
number of records in log file
        BOOL
        bLogSorted;
        int
        iFileSize;
// file size in bytes

// driver engine that
created this log file
        char
        szDriverEngineName[DRIVER_NAM
E_LEN];
// the record map
provides a fast way to get close to a particular
timestamp in a sorted log file.
//
        struct

```

```

//
{
        JULIAN_TIME
        TS;
// timestamp of record
//
        int
        iPos; // byte
position in file
}
RecMap[RecMapSize];
#define RecMapSize
200
} TXN_LOG_HEADER,
*PTXN_LOG_HEADER;

/* Header of the sorted pointers
blocks in Temp file (in merging). */
typedef struct BLOCK_HEADER {
        long
        BlockPos;
        __int64 CurPos;
        DWORD
        BytesRead;
        int
        nRecords;
        BYTE *offset;
// offset of pointers to records in
the log file */
} BLOCK_HEADER,
*PBLOCK_HEADER;

#define READ_BUFFER_SIZE
64*1024
#define WRITE_BUFFER_SIZE
8*1024
#define WRITE_BUFFER_SIZE
128*1024

#define NUM_READ_BUFFERS
1
#define NUM_WRITE_BUFFERS
2
#define MAX_NUM_BUFFERS
2

// flags passed in to the constructor
#define TXN_LOG_WRITE
0x01
#define TXN_LOG_READ
0x02
#define TXN_LOG_SORTED
0x04
#define TXN_LOG_CRASHOPEN 0x08
// if set, invalid
headers will be tolerated; used for recovery

#define TXN_LOG_OS_ERROR 1
#define TXN_LOG_NOT_SORTED 2

#define SKIP_CTRL_RECS 1

class CTxnLog
{
private:
        DWORD
        iBufferSize;
//buffer allocated size
        DWORD
        iBytesFreeInBuffer;
//total bytes available for use in
        buffer

```

```

int
iNumBuffers;
//buffers in use
int
iActiveBuffer;
//indicates which
buffer is active: 0 or 1
int
iIoBuffer;
//buffer
for any pending IO operation
//
int
iFilePointer;
//position in file.
LARGE_INTEGER
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
lastTS;
//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 in
the // last written buffer,
as the last record may be only partially in the
buffer.
// So remember the
timestamps for 2 last records that begin in the
buffer - one of
// them will the last
complete record written to disk.
JULIAN_TIME
PrevEndTxnTS;
// timestamp of the
previous to last record
union {
TXN_LOG_HEADER
HeaderForCheckpoint;// header
written on every checkpoint
char
szHeaderBuffer[512]; // 512
bytes is the minimum we can write to the disk
} HeaderBuffer;
//need the union because can't
write sizeof(TXN_LOG_HEADER) - too few bytes
// Control record
returned from GetNextRecord if the file
// currently opened
for read was not properly shut down
struct
{
TXN_RECORD_CONTROL
RecHeader;
char
szDriverName[DRIVER_NAME_LEN]
;
}
IncorrectShutDownRec;
BYTE
*pCurrent;
//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, LPCTSTR 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 <sqldb.h>

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

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_dblib.h"

#define DEFCLPACKSIZE
4096

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

const
        iMaxRetries = 10;
        // how many retries on deadlock
static long iConnectionCount = 0;
// number of current dblib connections

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout
expired";

BOOL APIENTRY DllMain(HMODULE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved)
{
        switch( ul_reason_for_call )
        {
                case
DLL_PROCESS_ATTACH:

                DisableThreadLibraryCalls(hModule)
;

                dbinit();
                // initialize dblib
                break;

                case
DLL_PROCESS_DETACH:

                dbexit();
                // close all dblib
                structures/connections

                break;

                default:

```



```

nothing */;
}
return TRUE;
}

int err_handler(DBPROCESS *dbproc, int severity,
int dberr, int oserr, LPCSTR dberrstr, LPCSTR
oserrstr)
{
    CTPCC_DBLIB
    *pConn;

    assert(dbproc != NULL);
    pConn =
(CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn-
>SetDbLibError( severity, dberr, oserr, dberrstr,
oserrstr );
    }
    return INT_CANCEL;
}

/* FUNCTION: int msg_handler(DBPROCESS
*dbproc, DBINT msgno, int msgstate, int
severity, char *msgtext)
*
* PURPOSE: This function handles DB-Library
SQL Server error messages
*
* ARGUMENTS:      DBPROCESS
                  *dbproc
                  DBPROCESS id pointer
*
                  DBINT
                  msgno
                  message number
*
                  int
                  msgstate
                  message state
*
                  int
                  severity
                  message severity
*
                  char
                  *msgtext          printable
message description
*
* RETURNS:      int

                  INT_CONTINUE      continue
if error is SQLETIME else INT_CANCEL action
*
                  INT_CANCEL
cancel operation
*
* COMMENTS:      This function also sets
the dead lock dbproc variable if necessary.
*
*/

// typedef INT (SQLAPI
*DBMSGHANDLE_PROC)(PDBPROCESS, DBINT,
INT, INT, LPCSTR, LPCSTR, LPCSTR,
DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT
msgno, int msgstate, int severity,

```

```

LPCSTR msgtext, LPCSTR srvname,
(LPCSTR procname, DBUSMALLINT line)
CTPCC_DBLIB
    *pConn;

    assert(dbproc != NULL);
    pConn =
(CTPCC_DBLIB*)dbgetuserdata(dbproc);

    if (pConn != NULL)
    {
        pConn->SetSqlError(
msgno, msgstate, severity, msgtext );
    }
    return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char
* pSrc, int n)
*
* PURPOSE: This function copies n characters
from string pSrc to pDst and places a
null
character at the end of the destination string.
*
* ARGUMENTS:      char
                  *pDest
                  destination string pointer
*
                  char
                  *pSrc      source string pointer
*
                  int
                  n
                  number of characters to copy
*
* RETURNS:      None
*
* COMMENTS:      Unlike strncpy this
function ensures that the result string is
always null terminated.
*/

inline static void UtilStrCpy(char * pDest, const
BYTE * pSrc, int n)
{
    strncpy(pDest, (char *)pSrc, n);
    pDest[n] = '\0';

    return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

char* CTPCC_DBLIB_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        {
            ERR_WRONG_SP_VERSION,
            "Wrong version of stored procs on
database server"
        },
    }
}

```

```

    {
        ERR_INVALID_CUST,
        "Invalid Customer id,name."
    },
    {
        ERR_NO_SUCH_ORDER,
        "No
orders found for customer."
    },
    {
        ERR_RETRIED_TRANS,
        "Retries
before transaction succeeded."
    },
    { 0,
        ""
    }
};

static char szNotFound[] =
"Unknown error number.";

for(i=0; errorMsgs[i].szMsg[0];
i++)
{
    if ( m_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::eDbSetMa
        xProcs);
    }

    // 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::eDbProch
        andler);

    if (dbprocmsghandle(login,
    msg_handler) == NULL)

        ThrowError(CDBLIBERR::eDbProch
        andler);

    DBSETUSER(login, szUser);
    DBSETLPWD(login, szPassword);
    DBSETHOST(login, szHost);
    DBSETPACKET(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 right database
        if (dbuse(m_dbproc, szDatabase)
        == FAIL)

            ThrowError(CDBLIBERR::eDbUse);

        dbcmd(m_dbproc, "set nocount on
        ");
        return row counts
        dbcmd(m_dbproc, "set
        XACT_ABORT ON");
        //
        rollback transaction on abort

        if (dbsqlexec(m_dbproc) == FAIL)

            ThrowError(CDBLIBERR::eDbSqlExe
            c);

        DiscardNextResults(2);

        // verify that version of stored procs
        on server is correct
        dbrcpinit(m_dbproc, "tpcc_version",
        0);

        if (dbrcpexec(m_dbproc) == FAIL)

            ThrowError(CDBLIBERR::eDbRpcEx
            ec);

        if (dbresults(m_dbproc) !=
        SUCCEEDED)

            ThrowError(CDBLIBERR::eDbResult
            s);

        if (dbnextrow(m_dbproc) !=
        REG_ROW)

            ThrowError(CDBLIBERR::eDbNextR
            ow);

        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

```

```

        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 DLib 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::eDbNextR
    ow);
            else
                break;
        }
        iRowsRead++;
    }

    if ((iExpectedCount >= 0) &&
        (iExpectedCount !=
    iRowsRead))
        ThrowError(CDBLIBERR::eWrongRo
    wCount);
}

// 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::eDbResult
    s);
            else
                break;
        }
        DiscardNextRows(-1);
        iResultsRead++;
    }

    if ((iExpectedCount >= 0) &&
        (iExpectedCount !=
    iResultsRead))
        ThrowError(CDBLIBERR::eWrongRo
    wCount);
}

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
        }
        catch (CDBLIBERR *e)
        {
            if (e->m_msgno == 1205 ||
                (e->m_msgno ==
                iErrOleDbProvider &&
                strstr(e->m_msgtext,
                sErrTimeoutExpired) != NULL) &&
                (++iTryCount <= iMaxRetries))
            {
                // hit deadlock; backoff for
                increasingly longer period
                delete e;
                Sleep(10 * iTryCount);
            }
            else
                throw;
        }
    } // while (TRUE)

    //if (iTryCount)
    //    throw new
    CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
    TRIED_TRANS, iTryCount);
}

void CTPCC_DBLIB::NewOrder()
{
    int
    i;
    DBINT
    commit_flag;
    DBDATETIME datetime;
    DBDATERECD daterec;
}

```

```

        if
        (dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcEx
    ec);

        if
        (dbresults(m_dbproc) != SUCCEEDED)
            ThrowError(CDBLIBERR::eDbResult
    s);

        if
        (dbnextrow(m_dbproc) != REG_ROW)
            ThrowError(CDBLIBERR::eDbNextR
    ow);

        if
        (pData=dbdata(m_dbproc, 1))
            m_txn.StockLevel.low_stock =
            *((long *) pData);

        DiscardNextRows(0);
        DiscardNextResults(0);

        m_txn.StockLevel.exec_status_code
        = eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e-
        >m_msgno == 1205 ||
            (e->m_msgno ==
            iErrOleDbProvider &&
            strstr(e->m_msgtext,
            sErrTimeoutExpired) != NULL) &&
            (++iTryCount <= iMaxRetries))
        {
            // hit deadlock; backoff for
            increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    } // while (TRUE)

    //if (iTryCount)
    //    throw new
    CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
    TRIED_TRANS, iTryCount);
}

void CTPCC_DBLIB::NewOrder()
{
    int
    i;
    DBINT
    commit_flag;
    DBDATETIME datetime;
    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_txn.NewOrder.w_id);

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.d_id);

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.c_id);

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

// check
whether any order lines are for a remote
warehouse

        m_txn.NewOrder.o_all_local = 1;
        for (i =
0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
            {
                m_txn.NewOrder.o_all_local = 0;
// at least one remote warehouse

                break;
            }
        }

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

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

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

```

```

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

        if
(dbrpcexec(m_dbproc) == FAIL)
            ThrowError(CDBLIBERR::eDbRpcEx
ec);

// Get
order line results

        m_txn.NewOrder.total_amount = 0;
        for (i =
0; i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            if (dbresults(m_dbproc) !=
SUCCEED)
                ThrowError(CDBLIBERR::eDbResult
s);

            if (dbnumcols(m_dbproc) != 5)
                ThrowError(CDBLIBERR::eWrongNu
mCols);

            if (dbnextrow(m_dbproc) !=
REG_ROW)
                ThrowError(CDBLIBERR::eDbNextR
ow);

            if(pData=dbdata(m_dbproc, 1))

                UtilStrCpy(m_txn.NewOrder.OL[i].ol
_i_name, pData, dbdatlen(m_dbproc, 1));

            if(pData=dbdata(m_dbproc, 2))

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

            if(pData=dbdata(m_dbproc, 3))

                UtilStrCpy(m_txn.NewOrder.OL[i].ol
_brand_generic, pData, dbdatlen(m_dbproc, 3));

            if(pData=dbdata(m_dbproc, 4))

                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_i_price, 8);

```

```

            if(pData=dbdata(m_dbproc, 5))

                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5),
SQLFLT8, (BYTE
*)&m_txn.NewOrder.OL[i].ol_amount, 8);

            m_txn.NewOrder.total_amount =
m_txn.NewOrder.total_amount +
m_txn.NewOrder.OL[i].ol_amount;

            DiscardNextRows(0);
        }

// get
remaining values for w_tax, d_tax, o_id, c_last,
c_discount, c_credit, o_entry_d, commit_flag
        if
(dbresults(m_dbproc) != SUCCEED)
            ThrowError(CDBLIBERR::eDbResult
s);

            if
(dbnextrow(m_dbproc) != REG_ROW)
                ThrowError(CDBLIBERR::eDbNextR
ow);

            if
(dbnumcols(m_dbproc) != 8)
                ThrowError(CDBLIBERR::eWrongNu
mCols);

            if
(pData=dbdata(m_dbproc, 1))

                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,1), SQLFLT8, (BYTE
*)&m_txn.NewOrder.w_tax, 8);

            if
(pData=dbdata(m_dbproc, 2))

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

            if
(pData=dbdata(m_dbproc, 3))

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

            if
(pData=dbdata(m_dbproc, 4))

```

```

        UtilStrCpy(m_txn.NewOrder.c_last,
pData, dbdatlen(m_dbproc, 4));
        if
(pData=dbdata(m_dbproc, 5))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);
        if
(pData=dbdata(m_dbproc, 6))
            UtilStrCpy(m_txn.NewOrder.c_cred
it, pData, dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))
            {
                datetime = *((DBDATETIME *)
pData);
                dbdatecrack(m_dbproc, &daterec,
&datetime);
                m_txn.NewOrder.o_entry_d.year
= daterec.year;
                m_txn.NewOrder.o_entry_d.month
= daterec.month;
                m_txn.NewOrder.o_entry_d.day
= daterec.day;
                m_txn.NewOrder.o_entry_d.hour
= daterec.hour;
                m_txn.NewOrder.o_entry_d.minute
= daterec.minute;
                m_txn.NewOrder.o_entry_d.second
= daterec.second;
            }
        if
(pData=dbdata(m_dbproc, 8))
            pData);
            commit_flag = *(DBTINYINT *)
pData);
            DiscardNextRows(0);
            DiscardNextResults(0);
            if
(commit_flag == 1)
                {
                    m_txn.NewOrder.total_amount *=
((1 + m_txn.NewOrder.w_tax +
m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));
                    m_txn.NewOrder.exec_status_code
= eOK;
                }
            else
                m_txn.NewOrder.exec_status_code
= eInvalidItem;
            return;
        }
    } catch (CSQLERR *e)

```

```

        {
            if ((e-
>m_msgno == 1205 ||
                (e->m_msgno ==
iErrOleDbProvider &&
                    strstr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
                (++iTryCount <= iMaxRetries))
                {
                    // hit deadlock; backoff for
increasingly longer period
                    delete e;
                    Sleep(10 * iTryCount);
                }
            else
                throw;
        }
    } // while (TRUE)
    // if (iTryCount)
    // throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
TRIED_TRANS, iTryCount);
}
void CTPCC_DBLIB::Payment()
{
    DBDATETIMEdatetime;
    DBDATEREC daterec;
    int
iTryCount = 0;
    const BYTE *pData;
    ResetError();
    while (TRUE)
    {
        try
        {
            dbrpcinit(m_dbproc,
"tpcc_payment", 0);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.w_id);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.c_w_id);
            dbrpcparam(m_dbproc, NULL, 0,
SQLFLT8, -1, -1, (BYTE *)
&m_txn.Payment.h_amount);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.d_id);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.Payment.c_d_id);
            dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.Payment.c_id);

```

```

            // if
customer id is zero, then payment is by name
            if
(m_txn.Payment.c_id == 0)
                dbrpcparam(m_dbproc, NULL, 0,
SQLCHAR, -1, strlen(m_txn.Payment.c_last),
(unsigned char *)m_txn.Payment.c_last);
            if
(dbrpcexec(m_dbproc) == FAIL)
                ThrowError(CDBLIBERR::eDbRpcEx
ec);
            if
(dbresults(m_dbproc) != SUCCEEDED)
                ThrowError(CDBLIBERR::eDbResult
s);
            if
(dbnxtrow(m_dbproc) != REG_ROW)
                ThrowError(CDBLIBERR::eDbNextR
ow);
            if
(dbnumcols(m_dbproc) != 27)
                ThrowError(CDBLIBERR::eWrongNu
mCols);
            if
(pData=dbdata(m_dbproc, 1))
                m_txn.Payment.c_id = *((DBINT *)
pData);
            if
(pData=dbdata(m_dbproc, 2))
                UtilStrCpy(m_txn.Payment.c_last,
pData, dbdatlen(m_dbproc, 2));
            if
(pData=dbdata(m_dbproc, 3))
                {
                    datetime = *((DBDATETIME *)
pData);
                    dbdatecrack(m_dbproc, &daterec,
&datetime);
                    m_txn.Payment.h_date.year =
daterec.year;
                    m_txn.Payment.h_date.month =
daterec.month;
                    m_txn.Payment.h_date.day =
daterec.day;
                    m_txn.Payment.h_date.hour =
daterec.hour;
                    m_txn.Payment.h_date.minute =
daterec.minute;
                    m_txn.Payment.h_date.second =
daterec.second;
                }
            if
(pData=dbdata(m_dbproc, 4))

```

```

        UtilStrCpy(m_txn.Payment.w_street
_1, pData, dbdatlen(m_dbproc, 4));
        if
(pData=dbdata(m_dbproc, 5))

        UtilStrCpy(m_txn.Payment.w_street
_2, pData, dbdatlen(m_dbproc, 5));
        if
(pData=dbdata(m_dbproc, 6))

        UtilStrCpy(m_txn.Payment.w_city,
pData, dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))

        UtilStrCpy(m_txn.Payment.w_state,
pData, dbdatlen(m_dbproc, 7));
        if
(pData=dbdata(m_dbproc, 8))

        UtilStrCpy(m_txn.Payment.w_zip,
pData, dbdatlen(m_dbproc, 8));
        if
(pData=dbdata(m_dbproc, 9))

        UtilStrCpy(m_txn.Payment.d_street
_1, pData, dbdatlen(m_dbproc, 9));
        if
(pData=dbdata(m_dbproc, 10))

        UtilStrCpy(m_txn.Payment.d_street
_2, pData, dbdatlen(m_dbproc, 10));
        if
(pData=dbdata(m_dbproc, 11))

        UtilStrCpy(m_txn.Payment.d_city,
pData, dbdatlen(m_dbproc, 11));
        if
(pData=dbdata(m_dbproc, 12))

        UtilStrCpy(m_txn.Payment.d_state,
pData, dbdatlen(m_dbproc, 12));
        if
(pData=dbdata(m_dbproc, 13))

        UtilStrCpy(m_txn.Payment.d_zip,
pData, dbdatlen(m_dbproc, 13));
        if
(pData=dbdata(m_dbproc, 14))

        UtilStrCpy(m_txn.Payment.c_first,
pData, dbdatlen(m_dbproc, 14));
        if
(pData=dbdata(m_dbproc, 15))

        UtilStrCpy(m_txn.Payment.c_middle
, pData, dbdatlen(m_dbproc, 15));
        if
(pData=dbdata(m_dbproc, 16))

        UtilStrCpy(m_txn.Payment.c_street
_1, pData, dbdatlen(m_dbproc, 16));
        if
(pData=dbdata(m_dbproc, 17))

        UtilStrCpy(m_txn.Payment.c_street
_2, pData, dbdatlen(m_dbproc, 17));
        if
(pData=dbdata(m_dbproc, 18))

        UtilStrCpy(m_txn.Payment.c_city,
pData, dbdatlen(m_dbproc, 18));
        if
(pData=dbdata(m_dbproc, 19))

```

```

        UtilStrCpy(m_txn.Payment.c_state,
pData, dbdatlen(m_dbproc, 19));
        if
(pData=dbdata(m_dbproc, 20))

        UtilStrCpy(m_txn.Payment.c_zip,
pData, dbdatlen(m_dbproc, 20));
        if
(pData=dbdata(m_dbproc, 21))

        UtilStrCpy(m_txn.Payment.c_phone,
pData, dbdatlen(m_dbproc, 21));
        if
(pData=dbdata(m_dbproc, 22))
        {
            datetime = *((DBDATETIME *)
pData);
            dbdatecrack(m_dbproc, &daterec,
&datetime);
            m_txn.Payment.c_since.year =
daterec.year;
            m_txn.Payment.c_since.month =
daterec.month;
            m_txn.Payment.c_since.day =
daterec.day;
            m_txn.Payment.c_since.hour =
daterec.hour;
            m_txn.Payment.c_since.minute =
daterec.minute;
            m_txn.Payment.c_since.second =
daterec.second;
        }
        if(pData=dbdata(m_dbproc, 23))
            UtilStrCpy(m_txn.Payment.c_credit,
pData, dbdatlen(m_dbproc, 23));
        if(pData=dbdata(m_dbproc, 24))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if(pData=dbdata(m_dbproc, 25))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if(pData=dbdata(m_dbproc, 26))
            dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if(pData=dbdata(m_dbproc, 27))
            UtilStrCpy(m_txn.Payment.c_data,
pData, dbdatlen(m_dbproc, 27));

        DiscardNextRows(0);

```

```

        DiscardNextResults(0);
        if
(m_txn.Payment.c_id == 0)
            throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
        else
            m_txn.Payment.exec_status_code
= eOK;
        return;
    }
    catch (CSQLERR *e)
    {
        if ((e-
>m_msgno == 1205 ||
(e->m_msgno ==
iErrOleDbProvider &&
strchr(e->m_msgtext,
sErrTimeoutExpired) != NULL)) &&
(++iTryCount <= iMaxRetries))
        {
            // hit deadlock; backoff for
increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
        else
            throw;
    }
    // while (TRUE)
    // if (iTryCount
    // throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
TRIED_TRANS, iTryCount);
}

void CTPCC_DBLIB::OrderStatus()
{
    int
i;
DBDATETIMEdatetime;
DBDATERECDaterec;

    int
iTryCount = 0;
RETCODE rc;
const BYTE *pData;

    ResetError();

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

```

```

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.w_id);

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT1, -1, -1, (BYTE *)
&m_txn.OrderStatus.d_id);

        dbrpcparam(m_dbproc, NULL, 0,
SQLINT4, -1, -1, (BYTE *)
&m_txn.OrderStatus.c_id);

// if
customer id is zero, then order status is by name
if
(m_txn.OrderStatus.c_id == 0)

        dbrpcparam(m_dbproc, NULL, 0,
SQLCHAR, -1, strlen(m_txn.OrderStatus.c_last),
(unsigned char *)m_txn.OrderStatus.c_last);

if
(dbrpcexec(m_dbproc) == FAIL)
        ThrowError(CDBLIBERR::eDbRpcEx
ec);

// Get
order lines
if
(dbresults(m_dbproc) != SUCCEED)
{
        if ((m_DbLibErr == NULL) &&
(m_SqlErr == NULL))

                throw new
CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );

        else

                ThrowError(CDBLIBERR::eDbResult
s);

if
(dbnumcols(m_dbproc) != 5)

        ThrowError(CDBLIBERR::eWrongNu
mCols);

        i = 0;
        while
(TRUE)
        {
                rc = dbnextrow(m_dbproc);

                if (rc == NO_MORE_ROWS)

                        break;

                if (rc != REG_ROW)

                        ThrowError(CDBLIBERR::eDbNextR
ow);

                if(pData=dbdata(m_dbproc, 1))

```

```

                m_txn.OrderStatus.OL[i].ol_supply_
w_id = (*(DBSMALLINT *) pData);
                if(pData=dbdata(m_dbproc, 2))

                        m_txn.OrderStatus.OL[i].ol_i_id =
(*(DBINT *) pData);

                if(pData=dbdata(m_dbproc, 3))

                        m_txn.OrderStatus.OL[i].ol_quantit
y = (*(DBSMALLINT *) pData);

                if(pData=dbdata(m_dbproc, 4))

                        dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,4),

                        SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);

                if(pData=dbdata(m_dbproc, 5))

                        {

                                datetime =
*((DBDATETIME *) pData);

                                dbdatecrack(m_dbproc, &daterec,
&datetime);

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.year = daterec.year;

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.month = daterec.month;

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.day = daterec.day;

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.hour = daterec.hour;

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.minute = daterec.minute;

                                m_txn.OrderStatus.OL[i].ol_delivery
_d.second = daterec.second;

                        }

                        i++;

                m_txn.OrderStatus.o_ol_cnt = i;

if
(dbresults(m_dbproc) != SUCCEED)

                ThrowError(CDBLIBERR::eDbResult
s);

```

```

if
(dbnextrow(m_dbproc) != REG_ROW)
        ThrowError(CDBLIBERR::eDbNextR
ow);

if
(dbnumcols(m_dbproc) != 8)

        ThrowError(CDBLIBERR::eWrongNu
mCols);

        if(pData=dbdata(m_dbproc, 1))

                m_txn.OrderStatus.c_id = (*(DBINT
*) pData);

        if(pData=dbdata(m_dbproc, 2))

                UtilStrCpy(m_txn.OrderStatus.c_last
, pData, dbdatlen(m_dbproc,2));

        if(pData=dbdata(m_dbproc, 3))

                UtilStrCpy(m_txn.OrderStatus.c_firs
t, pData, dbdatlen(m_dbproc,3));

        if(pData=dbdata(m_dbproc, 4))

                UtilStrCpy(m_txn.OrderStatus.c_mi
ddle, pData, dbdatlen(m_dbproc, 4));

        if(pData=dbdata(m_dbproc, 5))
        {
                datetime = *((DBDATETIME *)
pData);

                dbdatecrack(m_dbproc, &daterec,
&datetime);

                m_txn.OrderStatus.o_entry_d.year
= daterec.year;

                m_txn.OrderStatus.o_entry_d.mont
h = daterec.month;

                m_txn.OrderStatus.o_entry_d.day
= daterec.day;

                m_txn.OrderStatus.o_entry_d.hour
= daterec.hour;

                m_txn.OrderStatus.o_entry_d.minut
e = daterec.minute;

                m_txn.OrderStatus.o_entry_d.secon
d = daterec.second;

        }

        if(pData=dbdata(m_dbproc, 6))

                m_txn.OrderStatus.o_carrier_id =
(*(DBSMALLINT *) pData);

        if(pData=dbdata(m_dbproc, 7))

                dbconvert(m_dbproc,
SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,7),

                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);

```

```

        if(pData=dbdata(m_dbproc, 8))

            m_txn.OrderStatus.o_id =
            (*(DBINT *) pData);

            DiscardNextRows(0);

            DiscardNextResults(0);

            if
            (m_txn.OrderStatus.o_ol_cnt == 0)

                throw new CTPCC_DBLIB_ERR(
                CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
            else if
            (m_txn.OrderStatus.c_id == 0 &&
            m_txn.OrderStatus.c_last[0] == 0)

                throw new CTPCC_DBLIB_ERR(
                CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
            else

                m_txn.OrderStatus.exec_status_cod
e = eOK;

                return;
            }
            catch (CSQLERR *e)
            {
                if ((e-
                >m_msgno == 1205 ||

                (e->m_msgno ==
                iErrOleDbProvider &&

                strstr(e->m_msgtext,
                sErrTimeoutExpired) != NULL)) &&

                (++iTryCount <= iMaxRetries))
                {
                    // hit deadlock; backoff for
                    increasingly longer period

                    delete e;

                    Sleep(10 * iTryCount);
                }
                else

                    throw;
            }
            // while (TRUE)

            // if (iTryCount)
            // throw new
            CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
            TRIED_TRANS, iTryCount);
        }

        void CTPCC_DBLIB::Delivery()
        {
            int
            i;
            int
            iTryCount = 0;
            const BYTE *pData;

            ResetError();

            while (TRUE)

```

```

        {
            try
            {
                dbrpcinit(m_dbproc,
                "tpcc_delivery", 0);

                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT4, -1, -1, (BYTE *)
                &m_txn.Delivery.w_id);

                dbrpcparam(m_dbproc, NULL, 0,
                SQLINT1, -1, -1, (BYTE *)
                &m_txn.Delivery.o_carrier_id);

                if
                (dbrpcexec(m_dbproc) == FAIL)

                    ThrowError(CDBLIBERR::eDbRpcEx
                    ec);

                if
                (dbresults(m_dbproc) != SUCCEED)

                    ThrowError(CDBLIBERR::eDbResult
                    s);

                if
                (dbnextrow(m_dbproc) != REG_ROW)

                    ThrowError(CDBLIBERR::eDbNextR
                    ow);

                if
                (dbnumcols(m_dbproc) != 10)

                    ThrowError(CDBLIBERR::eWrongNu
                    mCols);

                for (i=0;
                i<10; i++)
                {
                    if (pData = dbdata(m_dbproc, i+1))

                        m_txn.Delivery.o_id[i]
                        = *((DBINT *)pData);

                    DiscardNextRows(0);

                    DiscardNextResults(0);

                    m_txn.Delivery.exec_status_code =
                    eOK;

                    return;
                }
                catch (CSQLERR *e)
                {
                    if ((e-
                    >m_msgno == 1205 ||

                    (e->m_msgno ==
                    iErrOleDbProvider &&

                    strstr(e->m_msgtext,
                    sErrTimeoutExpired) != NULL)) &&

                    (++iTryCount <= iMaxRetries))
                    {

```

```

                    // hit deadlock; backoff for
                    increasingly longer period
                    delete e;

                    Sleep(10 * iTryCount);
                }
            }
            else

                throw;
        }
        // while (TRUE)

        // if (iTryCount)
        // throw new
        CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RE
        TRIED_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 dblink tpcc_dblink.h

```

/* FILE:
TPCC_DBLIB.H

* Microsoft TPC-C Kit Ver. 4.20.000

* Copyright Microsoft, 1999
* All Rights Reserved

*
* Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
*
* PURPOSE: Header file for TPC-C
txn class implementation.
*
* Change history:
* 4.20.000 - updated
rev number to match kit
*/
#pragma once

#ifndef PDBPROCESS
#define DBPROCESS void // dbprocess
structure type
typedef DBPROCESS * PDBPROCESS;
#endif

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.

```



```

#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
public:
    CSQLERR(void)
    {
        m_msgno = 0;
        m_msgstate = 0;
        m_severity = 0;
        m_msgtext = NULL;
    };
    ~CSQLERR()
    {
        delete []
        m_msgtext;
    };
    int
    m_msgno;
    int
    m_msgstate;
    int
    m_severity;
    char *m_msgtext;

    int
    ErrorType() {return
ERR_TYPE_SQL;};
    char*
    ErrorTypeStr() { return "SQL"; }
    int
    ErrorNum() {return m_msgno;};
    char*
    ErrorText() {return m_msgtext;};
};

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,
        // error from dblogin
        eDbOpen,
        // error from dbopen
        eDbUse,
        // error from dbuse
        eDbSqlExec,
        // error from dbsqlexec
        eDbSet,
        // error from one of the dbset*
        routines
        eDbNextRow,
        // error from dbnextrow
    };

```

```

        eWrongRowCount,
        // more or less rows returned than
        expected
        eWrongNumCols,
        // more or less columns returned
        than expected
        eDbResults,
        // error from dbresults
        eDbRpcExec,
        // error from dbrpcexec
        eDbSetMaxProcs,
        // error from dbsetmaxprocs
        eDbProcHandler
        // error from either dbprocerrhandle
        or dbprocmsgghandle
    };
    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...
    PDBPROCESS
    m_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 <sqlext.h>

// #define COMPILE_FOR_SNAC // define
that to compile for SQL Native Client; comment
out to use MDAC

#ifndef COMPILE_FOR_SNAC
#include <odbcss.h>
#else
// Compile for SNAC
#include <sqlncli.h>
#endif

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

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_odbcc.h"

```

```

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.20.000";

const iMaxRetries = 10; // how
many retries on deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout
expired";

static SQLHENV henv = SQL_NULL_HENV;

// ODBC environment handle

BOOL APIENTRY DllMain(HMODULE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved)
{
        switch( ul_reason_for_call )
        {
                case
DLL_PROCESS_ATTACH:

                DisableThreadLibraryCalls(hModule)
;

                if (
SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )

                return FALSE;

                break;

                case
DLL_PROCESS_DETACH:

                if (henv
!= NULL)

                SQLFreeEnv(henv);

                break;

                default:
                /*
nothing */;
        }
        return TRUE;
}

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
*
*/

char* CTPCC_ODBC_ERR::ErrorText(void)
{
        int i;

        static SERRORMSG errorMsgs[] =
        {
                {
ERR_WRONG_SP_VERSION,
"Wrong version of stored procs on
database server"
},
                {
ERR_INVALID_CUST,
"Invalid Customer id.name."
},
                {
ERR_NO_SUCH_ORDER,
"No
orders found for customer."
}
        }
}

```

```

        {
ERR_RETRIED_TRANS,          "Retries
before transaction succeeded."
        },
        { 0,
        ""
        }
};

static char szNotFound[] =
"Unknown error number.";

for(i=0; errorMsgs[i].szMsg[0];
i++)
{
    if ( m_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(bCallNoDuplicates
NewOrder)
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

    m_hstmtNewOrder =
SQL_NULL_HSTMT;
    m_hstmtPayment =
SQL_NULL_HSTMT;
    m_hstmtDelivery =
SQL_NULL_HSTMT;
    m_hstmtOrderStatus =
SQL_NULL_HSTMT;
    m_hstmtStockLevel =
SQL_NULL_HSTMT;

    m_descNewOrderCols1 =
SQL_NULL_HDESC;
    m_descNewOrderCols2 =
SQL_NULL_HDESC;
    m_descOrderStatusCols1 =
SQL_NULL_HDESC;
    m_descOrderStatusCols2 =
SQL_NULL_HDESC;

    wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));

    if (
SQLAllocHandle(SQL_HANDLE_DBC, henv,
&m_hdbc) != SQL_SUCCESS )

        ThrowError(CODBCERR::eAllocHand
le);

    if ( SQLSetConnectOption(m_hdbc,
SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )

        ThrowError(CODBCERR::eConnOpti
on);

    {
        char
szConnectStr[256];
        char
szOutStr[1024];
        SQLSMALLINT
iOutStrLen;

#ifdef COMPILE_FOR_SNAC
        sprintf( szConnectStr,
"DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABA
SE=%s",
szServer, szUser, szPassword,
szDatabase );
#else
        // Compile for SNAC
        sprintf( szConnectStr,
"DRIVER=SQL Native
Client;SERVER=%s;UID=%s;PWD=%s;DATABA
E=%s",

```

```

szServer, szUser, szPassword,
szDatabase );
        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::eAllocHand
le);

    {
        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::eExecDirec
t);

        // 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::eExecDirec
t);

        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_STM
T, 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_STM
T, m_hstmtNewOrder);
    SQLFreeHandle(SQL_HANDLE_STM
T, m_hstmtPayment);
    SQLFreeHandle(SQL_HANDLE_STM
T, m_hstmtDelivery);
    SQLFreeHandle(SQL_HANDLE_STM
T, m_hstmtOrderStatus);
    SQLFreeHandle(SQL_HANDLE_STM
T, m_hstmtStockLevel);

    SQLDisconnect(m_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC,
m_hdbc);
}

void CTPCC_ODBC::ThrowError(
COBDCERR::ACTION eAction )
{
    RETCODE          rc;
    SDWORD
    INativeError;
    char
    szState[6];
    char
    szMsg[SQL_MAX_MESSAGE_LEN
T];
    char
    szTmp[6*SQL_MAX_MESSAGE_LEN
GTH];
    COBDCERR *pODBCErr;
    // not allocated until
needed (maybe never)

    pODBCErr = new COBDCERR();

    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,
&INativeError,
        (BYTE *)&szMsg,
sizeof(szMsg), NULL);
        if (rc ==
SQL_NO_DATA)
            break;

        // check for deadlock

```

```

        if (INativeError ==
1205 || (INativeError == iErrOleDbProvider &&
strstr(szMsg, sErrTimeoutExpired)
!= NULL))
            pODBCErr->m_bDeadLock = TRUE;

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

        // 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(COBDCERR::eAllocHand
le);

    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(COBDCERR::eBindPara
m);

    if ( SQLBindCol(m_hstmt, 1,
SQL_C_SLONG, &m_txn.StockLevel.low_stock, 0,
NULL) != SQL_SUCCESS )
        ThrowError(COBDCERR::eBindCol);

    //Compose Stock Level statement
    _snwprintf(m_szStockLevelComman
d,
sizeof(m_szStockLevelCommand)/sizeof(m_szSto
ckLevelCommand[0]),
        L"{call
%stpcc_stocklevel(?,?,?)", m_szSPPrefix);
}

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

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt,
m_szStockLevelCommand, SQL_NTS);
            if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(COBDCERR::eExecDirec
t);

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

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            m_txn.StockLevel.exec_status_code
= eOK;
            break;
        }
        catch (COBDCERR *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_RE
TRIED_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::eAllocHand
le);

    m_hstmt = m_hstmtNewOrder;

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAt
tr);

    int i = 0;
    if ( SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.NewOrder.w_id, 0,
NULL) != SQL_SUCCESS
        ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0, &m_txn.NewOrder.d_id, 0,
NULL) != SQL_SUCCESS
        ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.NewOrder.c_id, 0,
NULL) != SQL_SUCCESS
        ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0, &m_txn.NewOrder.o_ol_cnt,
0, NULL) != SQL_SUCCESS
        ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0,
&m_txn.NewOrder.o_all_local, 0, NULL) !=
SQL_SUCCESS

```

```

)
ThrowError(CODBCERR::eBindPara
m);

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::eBindPara
m);

    // set the bind offset pointer
    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_ROW_BIND_OFFSET_PTR,
&m_BindOffset, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAt
tr);

    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::eSetStmtAt
tr);

    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_szNewOrderComman
d,
sizeof(m_szNewOrderCommand)/sizeof(m_szNew
OrderCommand[0]),
        // 0 1 2
        //
        012345678901234567890123456789
        L"{call
%stpc_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 ( SQLSetStmAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmAttr);

        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_of_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::eBindParameter);

        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::eBindParameter);
        }

        // set row-wise binding
        if ( SQLSetStmAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.NewOrder.OL[0]),
SQL_IS_UIINTEGER) != SQL_SUCCESS

```

```

        ||
SQLSetStmAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmAttr);

        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 ( SQLSetStmAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmAttr);

        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_szNewOrderNoDuplic
atesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/siz
eof(m_szNewOrderNoDuplicatesCommand[0]),
L"{call
%stpc_neworder_new(?,?,?,?,?,?,?,?,?,
?,?,?,?,?,?)",
L"?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?}");

        m_iBeginNewOrderNoDuplicatesVari
ablePart = 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_of_cnt; ++i)
    {
        for (j = i+1; j <
m_txn.NewOrder.o_of_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;

    // 0 1 2

    //
    012345678901234567890123456789
    wchar_t

    szSqlTemplate[iMAX_SP_NAME_LEN
];

L"{call tpcc_neworder(?,?,?,?" //
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_txn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L"}}" );

    // check whether any order lines
    are for a remote warehouse
    m_txn.NewOrder.o_all_local = 1;
    for ( i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
    {

```

```

        if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id) {
            m_txn.NewOrder.o_all_local = 0;
            // at least one remote warehouse
            break;
        }
    }
    while (TRUE)
    {
        try
        {
            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_txn.NewOrder.total_amount = 0;
            for ( i =
0; i < m_txn.NewOrder.o_ol_cnt; i++)
            {
                // set the bind offset value...
                m_BindOffset = i *
sizeof(m_txn.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_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
            }
            //
            associate the column bindings for the second
            result set
            if (
SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);
            if (
SQLFetch(m_hstmt) == SQL_ERROR)

```

```

                ThrowError(CODBCERR::eFetch);
                SQLFreeStmt(m_hstmt,
SQL_CLOSE);
            if
(m_no_commit_flag == 1)
            {
                m_txn.NewOrder.total_amount *=
((1 + m_txn.NewOrder.w_tax +
m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));
                m_txn.NewOrder.exec_status_code
= eOK;
            }
            else
                m_txn.NewOrder.exec_status_code
= eInvalidItem;
            break;
        }
        catch (CODBCERR *e)
        {
            if ((!e
> m_bDeadLock) || (++iTryCount >
iMaxRetries))
                throw;
            // hit
            deadlock; backoff for increasingly longer period
            delete e;
            Sleep(10
* iTryCount);
        }
    }
    //
    if (iTryCount)
    //
    // throw new
    CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RE
TRIED_TRANS, iTryCount);
}

//
// No lineitem duplicates optimized
//
// version.
//
void CTPCC_ODBC::NewOrderNoDuplicates()
{
    int i;
    RETCODE rc;
    int iTryCount = 0;

    // 0 1 2 3

    //
    0123456789012345678901234567890123
    wchar_t

    szSqlTemplate[iMAX_SP_NAME_LEN
];

```

```

L"{call tpcc_neworder_new(?,?,?,?," //
//
L"?,?,?,?,?,?," //
//
L"?,?,?,?,?,?," //
//
L"?,?,?,?,?,?");" //
//
m_hstmt =
m_hstmtNewOrderNoDuplicates;

// associate the parameter and
column bindings for this transaction
if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAt
tr);

// 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_txn.NewOrder.o_ol_cnt*6;
wcsncpy( &szSqlTemplate[i], L"");" );

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

while (TRUE)
{
try
{
//
configure block cursor
if (
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE, (SQLPOINTER)1,
0) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAt
tr);

rc =
SQLExecDirectW(m_hstmt, szSqlTemplate,
SQL_NTS);

```

```

if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
ThrowError(CODBCERR::eExecDirec
t);

configure block cursor
//
if
(SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_NEW_ORDER_ITEMS, 0)
!= SQL_SUCCESS)

ThrowError(CODBCERR::eSetStmtAt
tr);

// Get
order line results

if (
SQLFetch(m_hstmt) == SQL_ERROR)

ThrowError(CODBCERR::eFetch);

m_txn.NewOrder.total_amount = 0;
for ( i =
0; i < m_txn.NewOrder.o_ol_cnt; i++)
{
m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
}

//
associate the column bindings for the second
result set

if (
SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

ThrowError(CODBCERR::eSetStmtAt
tr);

// move
to the next resultset
if (
SQLMoreResults(m_hstmt) == SQL_ERROR )

ThrowError(CODBCERR::eMoreResu
lts);

if (
SQLFetch(m_hstmt) == SQL_ERROR)

ThrowError(CODBCERR::eFetch);

SQLFreeStmt(m_hstmt,
SQL_CLOSE);

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

```

```

m_txn.NewOrder.exec_status_code
= eOK;
}
else

m_txn.NewOrder.exec_status_code
= eInvalidItem;

break;
}
catch (CODBCERR *e)
{
if ((!e-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))

throw;

// hit
deadlock; backoff for increasingly longer period
delete e;
Sleep(10
* iTryCount);
}

//
if (iTryCount)
//
throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RE
TRIED_TRANS, iTryCount);
}

void CTPCC_ODBC::InitPaymentParams()
{
if (
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtPayment) != SQL_SUCCESS )

ThrowError(CODBCERR::eAllocHand
le);

m_hstmt = m_hstmtPayment;

int i = 0;
if ( SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.w_id, 0,
NULL) != SQL_SUCCESS

||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_w_id, 0,
NULL) != SQL_SUCCESS

||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE,
SQL_NUMERIC, 6, 2,
&m_txn.Payment.h_amount, 0, NULL) !=
SQL_SUCCESS

||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0,
NULL) != SQL_SUCCESS

||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT,
SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0,
NULL) != SQL_SUCCESS

||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0,
NULL) != SQL_SUCCESS

```



```

        ||
SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.Payment.c_last), 0,
&m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) !=
SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindPara
m);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.Payment.c_id,
0, NULL) !=
SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.h_date, 0, NULL)
!= SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_1,
sizeof(m_txn.Payment.w_street_1),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2,
sizeof(m_txn.Payment.w_street_2),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city,
sizeof(m_txn.Payment.w_city),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state,
sizeof(m_txn.Payment.w_state),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,
sizeof(m_txn.Payment.w_zip),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1,
sizeof(m_txn.Payment.d_street_1),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2,
sizeof(m_txn.Payment.d_street_2),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,
sizeof(m_txn.Payment.d_city),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,
sizeof(m_txn.Payment.d_state),
NULL) != SQL_SUCCESS

```

```

        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,
sizeof(m_txn.Payment.d_zip),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,
sizeof(m_txn.Payment.c_first),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,
sizeof(m_txn.Payment.c_middle),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1,
sizeof(m_txn.Payment.c_street_1),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2,
sizeof(m_txn.Payment.c_street_2),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,
sizeof(m_txn.Payment.c_city),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,
sizeof(m_txn.Payment.c_state),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,
sizeof(m_txn.Payment.c_zip),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_phone,
sizeof(m_txn.Payment.c_phone),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.c_since, 0, NULL)
!= SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_credit,
sizeof(m_txn.Payment.c_credit),
NULL) != SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_credit_lim, 0, NULL) !=
SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_discount, 0, NULL)
!= SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_balance, 0, NULL)
!= SQL_SUCCESS
        ||
SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_data,
sizeof(m_txn.Payment.c_data),
NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindCol);

```

```

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

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

    m_hstmt = m_hstmtPayment;
    if (m_txn.Payment.c_id != 0)
        m_txn.Payment.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt,
m_szPaymentCommand, SQL_NTS);
            if (rc !=
SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirec
t);

            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_RE
TRIED_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::eAllocHand
le);

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAt
tr);

    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::eBindPara
m);

    // 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::eSetStmtAt
tr);

    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::eSetStmtAt
tr);

    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_szOrderStatusComma
nd,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOr
derStatusCommand[0]), L"{call
%stppcc_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::eSetStmtAt
tr);

    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::eSetStmtAt
tr);

            rc =
SQLExecDirectW(m_hstmt,
m_szOrderStatusCommand, SQL_NTS);
            if ( ((rc
== SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0)) || (rc == SQL_ERROR) )
                ThrowError(CODBCERR::eExecDirec
t);

            //
            configure block cursor
            if (
SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS,
0) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAt
tr);

            rc =
SQLFetchScroll( m_hstmt, SQL_FETCH_NEXT, 0
);
            if ( ((rc
== SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0)) || (rc == SQL_ERROR) )
                ThrowError(CODBCERR::eFetchScri
ll);

```

```

        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_RE
TRIED_TRANS, iTryCount);

void CTPCC_ODBC::InitDeliveryParams()
{
    if (
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtDelivery) != SQL_SUCCESS )

        ThrowError(CODBCERR::eAllocHand
le);

        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::eBindPara
m);

        for (i=0;i<10;i++)
        {
            if (
SQLBindCol(m_hstmt, (UWORD)(i+1),
SQL_C_SLONG, &m_txn.Delivery.o_id[i], 0,
NULL) != SQL_SUCCESS )

                ThrowError(CODBCERR::eBindCol);
        }

        //Compose Delivery statement

        _snwprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDelive
ryCommand[0]),
            L"(call
%stpc_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::eExecDirec
t);

            if (
SQLFetch(m_hstmt) == SQL_ERROR )

```

```

        ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        m_txn.Delivery.exec_status_code =
eOK;

        break;
    }
    catch (CODBCERR *e)
    {
        if ((!e-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))

            throw;

            // hit
            deadlock; backoff for increasingly longer period
            delete e;
            Sleep(10

            * iTryCount);
    }
}

//
if (iTryCount)
    throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RE
TRIED_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.
#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

#define iMAX_SP_NAME_LEN 256
//maximum length of a stored
procedure name with parameters

class CODBCERR : public CBaseErr
{
public:
enum ACTION

```

```

        {
            eNone,

            eUnknown,

            eAllocConn,
            // error from SQLAllocConnect

            eAllocHandle, // error
            from SQLAllocHandle

            eConnOption,
            // error from SQLSetConnectOption

            eConnect,
            // error from SQLConnect

            eAllocStmt,
            // error from SQLAllocStmt

            eExecDirect, // error
            from SQLExecDirect

            eBindParam,
            // error from SQLBindParameter

            eBindCol,
            // error from SQLBindCol

            eFetch,

            // error from SQLFetch

            eFetchScroll, // error
            from SQLFetchScroll

            eMoreResults,
            // error from SQLMoreResults

            ePrepare,
            // error from SQLPrepare

            eExecute,
            // error from SQLExecute

            eSetEnvAttr, // error
            from SQLSetEnvAttr

            eSetStmtAttr // error
            from SQLSetStmtAttr
        };

        CODBCERR(void)
        {
            m_eAction = eNone;
            m_NativeError = 0;
            m_bDeadLock = FALSE;
            m_odbcerrstr = NULL;
        };

        ~CODBCERR()
        {
            if
            (m_odbcerrstr != NULL)

            delete [] m_odbcerrstr;
        };

        ACTION
        m_eAction;
        int
        m_NativeError;

```

```

        BOOL
        m_bDeadLock;char *m_odbcerrstr;

        int
        ErrorType() {return
        ERR_TYPE_ODBC;};

        char*
        ErrorTypeStr() { return "ODBC"; }

        int
        ErrorNum() {return
        m_NativeError;};

        char*
        ErrorText() {return m_odbcerrstr;};

        int
        ErrorAction() { return
        (int)m_eAction; }
    };

    class CTPCC_ODBC_ERR : public CBaseErr
    {
    public:
        enum
        TPCC_ODBC_ERRS
        {
            ERR_WRONG_SP_VERSION = 1,
            // "Wrong version of stored procs
            on database server"

            ERR_INVALID_CUST,
            // "Invalid Customer
            id,name."

            ERR_NO_SUCH_ORDER,
            // "No orders found
            for customer."

            ERR_RETRIED_TRANS,
            // "Retries before
            transaction succeeded."
        };

        CTPCC_ODBC_ERR(
        int iErr ) { m_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
        ErrorType() {return
        ERR_TYPE_TPCC_ODBC;};

        char*
        ErrorTypeStr() { return "TPCC
        ODBC"; }

        int
        ErrorNum() {return m_erno;};

        char*
        ErrorText();
    };

    class DIIDecl 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_szNewOrderNoDuplicatesComma
        nd[iMAX_SP_NAME_LEN];

        int

        m_iBeginNewOrderVariablePart;
        // beginning of the variable part in
        NewOrder statement

        int

        m_iBeginNewOrderNoDuplicatesVari
        ablePart; // beginning of the variable part in
        NewOrder statement

        wchar_t
        m_szPaymentCommand[iMAX_SP_N
        AME_LEN];

```

```

        wchar_t
        m_szDeliveryCommand[iMAX_SP_N
AME_LEN];
        wchar_t
        m_szOrderStatusCommand[iMAX_S
P_NAME_LEN];
        wchar_t
        m_szStockLevelCommand[iMAX_SP
_NAME_LEN];

        // new-order specific
fields
        SQLUIINTEGER
        m_BindOffset;
        SQLUIINTEGER
        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(
                LPCSTR szServer, LPCSTR szUser,
                LPCSTR szPassword,
                LPCSTR szHost,
                LPCSTR szDatabase,
                LPCWSTR szSPPrefix,
                BOOL bCallNoDuplicatesNewOrder);

```

```

        ~CTPCC_ODBC(void);
        inline
        PNEW_ORDER_DATA
        BuffAddr_NewOrder()
        { return &m_txn.NewOrder; };
        inline
        PPAYMENT_DATA
        BuffAddr_Payment()
        { return &m_txn.Payment; };
        inline
        PDELIVERY_DATA
        BuffAddr_Delivery()
        { return &m_txn.Delivery; };
        inline
        PSTOCK_LEVEL_DATA
        BuffAddr_StockLevel()
        { return &m_txn.StockLevel; };
        inline
        PORDER_STATUS_DATA
        BuffAddr_OrderStatus() { return
&m_txn.OrderStatus; };

        void NewOrder
        ();
        void Payment
        ();
        void Delivery
        ();
        void StockLevel
        ();
        void OrderStatus
        ();

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC*
CTPCC_ODBC_new
(
        LPCSTR szServer,
        LPCSTR szUser, LPCSTR szPassword,
        LPCSTR szHost,
        LPCSTR szDatabase,
        LPCWSTR szSPPrefix,
        BOOL bCallNoDuplicatesNewOrder );

typedef CTPCC_ODBC*
(TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCWSTR, BOOL);

tm_com_dll\src\tpcc_com.h

/* FILE:
TPCC_COM.H

* Microsoft TPC-C Kit Ver. 4.20.000

* Copyright Microsoft, 1999
* All Rights Reserved

* not yet audited

* PURPOSE: Header file for TPC-C
COM+ class implementation.
*
* Change history:
* 4.20.000 - first
version
*/

```

```

#pragma once
#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
        private:
                char
                m_szErrorText[64];

        public:
                // use this interface
                for genuine COM errors
                CCOMERR( HRESULT
                hr )
                {
                        m_hr =
                hr;
                        m_iErrorType = 0;
                        m_iError
                = 0;
                }

                // use this interface to
                impersonate a non-COM error type
                CCOMERR( int
                iErrorType, int iError )
                {
                        m_iErrorType = iErrorType;
                        m_iError
                = iError;
                        m_hr =
                S_OK;
                }

                int
                m_hr;
                int
                m_iErrorType;
                int
                m_iError;

                // A CCOMERR class
                can impersonate another class, which happens if
                the error
                // was not actually a
                COM Services error, but was simply transmitted
                back via COM.
                int ErrorType()
                {
                        if
                (m_iErrorType == 0)
                        return ERR_TYPE_COM;
                        else
                        return m_iErrorType;
                }

                char *ErrorTypeStr() {
                return "COM"; }

                int ErrorNum() {return
                m_hr;}

                char *ErrorText()

```

```

        {
            if (m_hr
== S_OK)
                sprintf( m_szErrorText, "Error:
Class %d, error # %d", m_iErrorType, m_iError
);
            else
                sprintf( m_szErrorText, "Error: COM
HRESULT %x", m_hr );
            return
m_szErrorText;
        }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

// COM Interface
pointers
    ITPCC*
    m_pNewOrder;
    ITPCC*
    m_pPayment;
    ITPCC*
    m_pStockLevel;
    ITPCC*
    m_pOrderStatus;

    struct COM_DATA
    {
        int
ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA
            NewOrder;

            PAYMENT_DATA
            Payment;

            DELIVERY_DATA
            Delivery;

            STOCK_LEVEL_DATA
            StockLevel;

            ORDER_STATUS_DATA
            OrderStatus;
        } u;
    } *m_pTxn;

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; };
PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel()
    { return &m_pTxn->u.StockLevel;
};

        inline
PORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return
&m_pTxn->u.OrderStatus; };

        void NewOrder
();
        void Payment
();
        void StockLevel
();
        void OrderStatus
();
        void Delivery
() { throw new
CCOMERR(E_NOTIMPL); } // not supported
};

    inline void ReleaseInterface(IUnknown *pUnk)
    {
        if (pUnk)
        {
            pUnk->Release();
            pUnk = NULL;
        }
    }

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM*
(TYPE_CTPCC_COM)(BOOL);

db_oledb_dll\tpcc_oledb.h

/* FILE:
TPCC_OLEDB.H
*
* Microsoft TPC-C Kit Ver. 4.20.000
*
* Copyright Microsoft, 1999-2004
*
* Written by Sergey Vasilevskiy
* All Rights Reserved
*
*
* PURPOSE: Header file for TPC-C
txn class OLE DB implementation.
*
*/
#pragma once

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.

```

```

#ifdef __declspec( dllimport )
#endif
#endif

#define IMAX_SP_NAME_LEN 256
//maximum length of a stored
procedure name with parameters

// Type of parameter and result set column
bindings.
enum eBindingType
{
    eInputParameter,
    eOutputParameter,
    eInputOutputParameter,
    eOutputColumn
};

class COLEDBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,

        eUnknown,

        eQueryInterface,
        // error from
QueryInterface

        eCreateSession,

        eCreateCommand,

        eSetCommandText,

        eExecute,
        // = 6

        eCreateAccessor,

        ePrepare,

        eGetNextRows,

        eGetData,

        eGetResult
        // = 11
    };

    COLEDBERR(LPCTSTR
szLoc)
    :
    CBaseErr(szLoc)
    {
        m_eAction = eNone;
        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_OLEDBErrStr = NULL;
    };

    ~COLEDBERR()
    {
        if
(m_OLEDBErrStr != NULL)
            delete [] m_OLEDBErrStr;
    };
};

```

```

        ACTION
    m_eAction;
    int
    m_NativeError;
    BOOL
    m_bDeadLock;
    char
    *m_OLEDBErrStr;

    int
    ErrorType() {return
ERR_TYPE_OLEDB;};
    char*
    ErrorTypeStr() { return "OLEDB"; }
    int
    ErrorNum() {return
m_NativeError;};
    char*
    ErrorText() {return
m_OLEDBErrStr;};
    int
    ErrorAction() { return
(int)m_eAction; }
};

class CTPCC_OLEDB_ERR : public CBaseErr
{
public:
    enum
    TPCC_OLEDB_ERRS
    {
        ERR_WRONG_SP_VERSION = 1,
        // "Wrong version of stored procs
on database server"

        ERR_INVALID_CUST,
        // "Invalid Customer
id,name."

        ERR_NO_SUCH_ORDER,
        // "No orders found
for customer."

        ERR_RETRIED_TRANS,
        // "Retries before
transaction succeeded."
    };

    CTPCC_OLEDB_ERR(
int iErr ) { m_errno = iErr; m_iTryCount = 0; };

    CTPCC_OLEDB_ERR(
int iErr, int iTryCount ) { m_errno = iErr;
m_iTryCount = iTryCount; };

    int
    m_errno;
    int
    m_iTryCount;

    int
    ErrorType() {return
ERR_TYPE_TPCC_OLEDB;};
    char*
    ErrorTypeStr() { return "TPCC
OLEDB"; }
    int
    ErrorNum() {return m_errno;};

    char*
    ErrorText();
};

```

```

class DllDecl CTPCC_OLEDB : public CTPCC_BASE
private:
    // declare variables
and private functions here...
    BOOL
    m_bDeadlock;
    // transaction was selected as
deadlock victim
    int
    m_MaxRetries;
    // retry count on
deadlock

    DBPROPSET
    m_rgInitPropSet;
    // initialization property set used to
establish a connection
    DBPROP
    m_InitProperties[4]; //
individual initialization properties

    IDBCreateSession*
    m_pIDBCreateSession; //
session (connection) interface

    IDBCreateCommand*
    m_pIDBCreateCommand; // SQL
command creation interface

    IMalloc*
    m_pIMalloc;
    // Needed to release
error strings.

    // StockLevel
ICommandText*
    m_pIStockLevelCommand;
    HACCESSOR
    m_hStockLevelInputAccessor;
    // accessor to bind input
parameters
    HACCESSOR
    m_hStockLevelOutputAccessor;
    // accessor to bind output columns
DBPARAMS
    m_StockLevelExecuteParams;
    // parameter structure
for Execute

    // NewOrder
// One prepared
command for each possible number of new order
line items
    ICommandText*
    m_pINewOrderCommand[MAX_OL_
NEW_ORDER_ITEMS];
    // accessors to bind
input parameters
    // one for each
possible number of new order line items
    HACCESSOR
    m_hNewOrderInputAccessor[MAX_
OL_NEW_ORDER_ITEMS];

```

```

    // accessor to bind
output columns of the first rowset
    HACCESSOR
    m_hNewOrderOutputAccessor[MAX_
OL_NEW_ORDER_ITEMS];
    // accessor to bind
output columns of the second rowset
    HACCESSOR
    m_hNewOrderOutputAccessor2[MA
X_OL_NEW_ORDER_ITEMS];
    // parameter structure
for Execute
    DBPARAMS
    m_NewOrderExecuteParams[MAX_
OL_NEW_ORDER_ITEMS];
    // Payment
ICommandText*
    m_pIPaymentCommand;
    HACCESSOR
    m_hPaymentInputAccessor;
    // accessor to bind input
parameters
    HACCESSOR
    m_hPaymentOutputAccessor;
    // accessor to bind output columns
DBPARAMS
    m_PaymentExecuteParams;
    // parameter structure
for Execute
    // OrderStatus
ICommandText*
    m_pIOrderStatusCommand;
    HACCESSOR
    m_hOrderStatusInputAccessor;
    // accessor to bind input
parameters
    HACCESSOR
    m_hOrderStatusOutputAccessor;
    // accessor to bind output columns
    HACCESSOR
    m_hOrderStatusOutputAccessor2;
    // accessor to bind output columns
DBPARAMS
    m_OrderStatusExecuteParams;
    // parameter structure
for Execute
    // Delivery
ICommandText*
    m_pIDeliveryCommand;
    HACCESSOR
    m_hDeliveryInputAccessor;
    // accessor to bind input
parameters
    HACCESSOR
    m_hDeliveryOutputAccessor;
    // accessor to bind output columns

```

```

DBPARAMS
    m_DeliveryExecuteParams;//
parameter structure for Execute

    wchar_t

    m_szSPPrefix[32];    // stored
procedures prefix

    // new-order specific
fields
    int

    m_no_commit_flag;

    void ThrowError(
IUnknown* pObjectWithError,
COLEDBERR::ACTION eAction, LPCTSTR
szLocation );

    void
CheckSPVersion();

    void
InitNewOrderParams();

    void
InitPaymentParams();

    void
InitDeliveryParams();

    void
InitStockLevelParams();

    void
InitOrderStatusParams();

    // Helper function to
create and prepare a command
    void
CreateCommand(wchar_t* szSQLCommand,
ICommandText** ppICommandText);
    // Helper function to
prepare a command
    void
PrepareCommand(ICommandText* pICommand);

    // Helper function to
fill one binding
    // Used for both input
parameter and output column bindings
    void
SetBinding(DBBINDING* pDBBinding, size_t
obValue, size_t cbMaxLen, DBTYPE wType);

    // Helper function to
initialize an array of bindings
    void
InitBindings(DBBINDING* pDBBindings, int
iCount, eBindingType BindingType);

    union
    {

        NEW_ORDER_DATA
        NewOrder;

        PAYMENT_DATA
        Payment;

        DELIVERY_DATA
        Delivery;

        STOCK_LEVEL_DATA
        StockLevel;

```

```

ORDER_STATUS_DATA
OrderStatus; }
    m_txn;

public:

    CTPCC_OLEDB(LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword, LPCSTR
szHost, LPCSTR szDatabase, LPCWSTR
szSPPrefix);

    ~CTPCC_OLEDB(void);

    inline
PNEW_ORDER_DATA
    BuffAddr_NewOrder()
    { return &m_txn.NewOrder; };
    inline
PPAYMENT_DATA
    BuffAddr_Payment()
    { return &m_txn.Payment; };
    inline
PDELIVERY_DATA
    BuffAddr_Delivery()
    { return &m_txn.Delivery; };
    inline
PSTOCK_LEVEL_DATA BuffAddr_StockLevel()
    { return &m_txn.StockLevel; };
    inline
PORDER_STATUS_DATA
    BuffAddr_OrderStatus() { return
&m_txn.OrderStatus; };

    void NewOrder
    ();
    void Payment
    ();
    void Delivery
    ();
    void StockLevel
    ();
    void OrderStatus
    ();

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_OLEDB*
CTPCC_OLEDB_new
    ( LPCSTR szServer, LPCSTR szUser,
LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase, LPCWSTR szSPPrefix );

typedef CTPCC_OLEDB*
(TYPE_CTPCC_OLEDB)(LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCWSTR);

db oledb dll\src\tpcc oledb
.cpp

/* FILE:
TPCC_OLEDB.CPP
*
Microsoft TPC-C Kit Ver. 4.42.000

```

```

*
* Copyright Microsoft, 2004
* Written by Sergey Vasilevskiy
* All Rights Reserved
*
*
*
* PURPOSE: Implements OLEDB
calls for TPC-C txns.
* Contact: Charles Levine
(clevine@microsoft.com)
*
*
*/

#include <windows.h>
#include <stdio.h>
#include <assert.h>
#include <stddef.h>

#define DBINITCONSTANTS
#include <oledb.h>
// #include <sqloledb.h> // Use MDAC
#include <sqlncli.h> // Use SNAC
#include <oledberr.h>

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

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_oledb.h"

#ifdef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

// version string; must match return value from
tpcc_version stored proc
const char sVersion[] = "4.42.000";

const iMaxRetries = 10; // how
many retries on deadlock

const int iErrOleDbProvider = 7312;
const char sErrTimeoutExpired[] = "Timeout
expired";

// this needs to be the same as the max length of
machine/database/user/password in Benchcraft
(engstut.h)
const static int iMaxNameLen = 32;

BOOL APIENTRY DllMain(HMODULE hModule,
DWORD ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case
DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule)
;
            break;

        case
DLL_PROCESS_DETACH:
            break;

```



```

        default:
            /*
nothing */;
        }
        return TRUE;
    }

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
*/
char* CTPCC_OLEDB_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        {
            ERR_WRONG_SP_VERSION,
            "Wrong version of stored procs on
database server"
        },
        {
            ERR_INVALID_CUST,
            "Invalid Customer id,name."
        },
        {
            ERR_NO_SUCH_ORDER,
            "No
orders found for customer."
        },
        {
            ERR_RETRIED_TRANS,
            "Retries
before transaction succeeded."
        },
        {
            0,
            ""
        }
    };

    static char szNotFound[] =
    "Unknown error number.";

    for(i=0; errorMsgs[i].szMsg[0];
i++)
    {
        if ( m_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_OLEDB*
CTPCC_OLEDB_new(
    LPCSTR szServer,
    // name of SQL server
    LPCSTR szUser,
    // user name for login
    LPCSTR szPassword,
    // password for login
    LPCSTR szHost,
    // not used
    LPCSTR szDatabase,
    // name of database to use

```

```

    LPCWSTR szSPPrefix )
    // prefix to append to the stored
procedure names
    return new CTPCC_OLEDB(
szServer, szUser, szPassword, szHost,
szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
    LPCSTR szServer,
    // name
of SQL server
    LPCSTR szUser,
    // user name for login
    LPCSTR szPassword,
    //
password for login
    LPCSTR szHost,
    // not used
    LPCSTR szDatabase,
    // name
of database to use
    LPCWSTR szSPPrefix
    // prefix
to append to the stored procedure names
)
: m_pIMalloc(NULL)
{
    int iRc;
    int hr;
    HRESULT hr;
    IDBInitialize*
pIDBInitialize = NULL;
    // data source interface
    IDBProperties*
pIDBProperties = NULL;
    ICommandText*
pICommandText;
    // SQL
command without parameters
    wchar_t
szwServer[iMaxNameLen];
    // Unicode string used to convert to
BSTR
    wchar_t
szwDatabase[iMaxNameLen];
    // Unicode string used to convert to
BSTR
    wchar_t
szwUser[iMaxNameLen];
    // Unicode string used to convert to
BSTR
    wchar_t
szwPassword[iMaxNameLen];
    // Unicode string used to convert to
BSTR
    // Copy stored procedures prefix
    wcsncpy(m_szSPPrefix, szSPPrefix,
sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));
    // Convert single byte ANSI strings
to Unicode (for later conversion to BSTR)

```

```

        iRc =
MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szServer,
(int)strlen(szServer)+1, szwServer,
iMaxNameLen);
        iRc =
MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szDatabase,
(int)strlen(szDatabase)+1, szwDatabase,
iMaxNameLen);
        iRc =
MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szUser,
(int)strlen(szUser)+1, szwUser, iMaxNameLen);
        iRc =
MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szPassword,
(int)strlen(szPassword)+1, szwPassword,
iMaxNameLen);

        // Initialize COM library to be able
to use OLE-DB interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB
component
        //hr =
CoCreateInstance(CLSID_SQLOLEDB, // GUID of
SQLOLEDB component
        // Compile for SNAC
        hr =
CoCreateInstance(CLSID_SQLNCLI, // GUID
of SQLNCLI component
        NULL,
        // not defining an
aggregate component, so NULL
        CLSCTX_INPROC_SERVER,
        // run the component in our
process
        IID_IDBInitialize,
        (void **) &pIDBInitialize);
    /*
Initialize the property values needed
to establish the connection.
*/
    for(i = 0; i < 4; i++)
        VariantInit(&m_InitProperties[i].vValue);
    //Server name.
    m_InitProperties[0].dwPropertyID =
DBPROP_INIT_DATASOURCE;
    m_InitProperties[0].vValue.vt = VT_BSTR;
    m_InitProperties[0].vValue.bstrVal=
SysAllocString(szwServer);
    m_InitProperties[0].dwOptions =
DBPROP_OPTIONS_REQUIRED;
    m_InitProperties[0].colid = DB_NULLID;
    //Database.
    m_InitProperties[1].dwPropertyID =
DBPROP_INIT_CATALOG;
    m_InitProperties[1].vValue.vt = VT_BSTR;
    m_InitProperties[1].vValue.bstrVal=
SysAllocString(szwDatabase);
    m_InitProperties[1].dwOptions =
DBPROP_OPTIONS_REQUIRED;
    m_InitProperties[1].colid = DB_NULLID;
    //Username (login).
    m_InitProperties[2].dwPropertyID =
DBPROP_AUTH_USERID;
    m_InitProperties[2].vValue.vt = VT_BSTR;
    m_InitProperties[2].vValue.bstrVal=
SysAllocString(szwUser);
    m_InitProperties[2].dwOptions =
DBPROP_OPTIONS_REQUIRED;
    m_InitProperties[2].colid = DB_NULLID;
    //Password.

```

```

    m_InitProperties[3].dwPropertyID =
    DBPROP_AUTH_PASSWORD;
    m_InitProperties[3].vValue.vt = VT_BSTR;
    m_InitProperties[3].vValue.bstrVal=
    SysAllocString(szwPassword);
    m_InitProperties[3].dwOptions =
    DBPROP_OPTIONS_REQUIRED;
    m_InitProperties[3].colid = DB_NULLID;
    /*
    Construct the DBPROPSET
    structure(m_rgInitPropSet). The
    DBPROPSET structure is used to pass an array
    of DBPROP
    structures (m_InitProperties) to the
    SetProperties method.
    */
    m_rgInitPropSet.guidPropertySet =
    DBPROPSET_DBINIT;
    m_rgInitPropSet.cProperties = 4;
    m_rgInitPropSet.rgProperties =
    m_InitProperties;
    //Set initialization properties.
    if (FAILED(hr = pIDBInitialize-
    >QueryInterface(IID_IDBProperties,
    (void
    ***)&pIDBProperties)))
    {
        ThrowError(pIDBInitialize,
        COLEDBERR::eQueryInterface,
        "CTPCC_OLEDB()");
    }

    hr = pIDBProperties->SetProperties(1,
    &m_rgInitPropSet);

    pIDBProperties->Release();
    //Now establish the connection to the data
    source.
    hr = pIDBInitialize->Initialize();

    // Free BSTR property strings
    for(i = 0; i < 4; i++)
    {
        SysFreeString(m_InitProperties[i].vValue.bstrVal)
    }

    hr = pIDBInitialize-
    >QueryInterface(IID_IDBCreateSession, (void
    ***)&m_pIDBCreateSession);

    // Releasing this has no effect on
    the SQL Server connection
    // of the data source object because of the
    reference maintained by
    // m_pIDBCreateSession.
    pIDBInitialize->Release();
    pIDBInitialize = NULL;

    hr = m_pIDBCreateSession-
    >CreateSession(NULL, IID_IDBCreateCommand,
    (IUnknown ***)&m_pIDBCreateCommand);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateSession,
        COLEDBERR::eCreateSession,
        "CTPCC_OLEDB()");
    }

```

```

    hr = m_pIDBCreateCommand-
    >CreateCommand(NULL, IID_IDBCreateCommand,
    (IUnknown ***)&m_pIDBCreateCommand);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,
        COLEDBERR::eCreateCommand,
        "CTPCC_OLEDB()");
    }

    hr = pIDBCreateCommand-
    >SetCommandText(DBGUID_SQL, L"set nocount
    on set XACT_ABORT ON");
    if (FAILED(hr))
    {
        ThrowError(pIDBCreateCommand,
        COLEDBERR::eSetCommandText,
        "CTPCC_OLEDB()");
    }

    hr = pIDBCreateCommand-
    >Execute(NULL, IID_NULL, NULL, NULL, NULL);
    if (FAILED(hr))
    {
        ThrowError(pIDBCreateCommand,
        COLEDBERR::eExecute, "CTPCC_OLEDB()");
    }

    pIDBCreateCommand->Release();

    // verify that version of stored procs
    on server is correct
    CheckSPVersion();

    // Get IMalloc interface
    hr = CoGetMalloc(1, (LPMAALLOC
    ***)&m_pIMalloc);

    // Bind parameters for each of the
    transactions
    InitNewOrderParams();
    InitPaymentParams();
    InitOrderStatusParams();
    InitDeliveryParams();
    InitStockLevelParams();
}

CTPCC_OLEDB::~~CTPCC_OLEDB( void )
{
    if (m_pIMalloc != NULL)
    {
        m_pIMalloc-
        >Release();
    }
    m_pIPaymentCommand-
    >Release();
    m_pIDBCreateCommand-
    >Release();
    m_pIDBCreateSession->Release();

    CoUninitialize(); //
    uninitialize COM library
}

/*
 * Check stored procedures version on
the server.
*/
void CTPCC_OLEDB::CheckSPVersion()
{
    HRESULT
    hr;

```

```

    char
    *pCommandText[10];
    pIDBCreateCommand,
    pIDBCreateCommand,
    IAccessor*
    pIAccessor;
    IRowset*
    pRowset;
    const ULONG
    nOutputParams = 1; // output
    1st result set columns
    HACCESSOR
    hTpccVersionOutputAccessor;
    // Structure to bind in accessor
    DBBINDING
    acOutputDBBinding[nOutputParams
    ];
    DBBINDSTATUS
    acOutputDBBindStatus[nOutputPara
    ms];
    LONG
    cRows = 1; // number of rows
    returned in the rowset
    ULONG
    cRowsObtained;
    HROW
    rghRow;
    //returned row handles
    HROW*
    prghRow = &rghRow;

    hr = m_pIDBCreateCommand-
    >CreateCommand(NULL, IID_IDBCreateCommand,
    (IUnknown ***)&pIDBCreateCommand);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,
        COLEDBERR::eCreateCommand,
        "CheckSPVersion()");
    }

    hr = pIDBCreateCommand-
    >SetCommandText(DBGUID_SQL, L"{call
    tpcc_version}");
    if (FAILED(hr))
    {
        ThrowError(pIDBCreateCommand,
        COLEDBERR::eSetCommandText,
        "CheckSPVersion()");
    }

    hr = pIDBCreateCommand-
    >QueryInterface(IID_IAccessor, (void
    ***)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(pIDBCreateCommand,
        COLEDBERR::eQueryInterface,
        "CheckSPVersion()");
    }

    // Now fill the binding information
    for result set 1 output columns
    InitBindings(&acOutputDBBinding[0
    ], nOutputParams, eOutputColumn);

    // Binding for a rowset
    SetBinding(&acOutputDBBinding[0],
    0, sizeof(db_sp_version), DBTYPE_STR);

    hr = pIAccessor->CreateAccessor(
    DBACCESSOR_ROWDATA,

```

```

        nOutputParams,
        acOutputDBBinding,
        sizeof(db_sp_version),
&hTpcVersionOutputAccessor,
        acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
        COLEDBERR::eCreateAccessor,
        "CheckSPVersion()");
    }

    hr = pICommandText-
>Execute(NULL, IID_IRowset, NULL, NULL,
(IUnknown **)&pRowset);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
        COLEDBERR::eExecute, "CheckSPVersion()");
    }

    // Fetch the result row handle(s)
    hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
        COLEDBERR::eGetNextRows,
        "CheckSPVersion()");
    }

    // Fetch the actual row data by
handle
    hr = pRowset->GetData(rghRow,
hTpcVersionOutputAccessor, &db_sp_version);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
        COLEDBERR::eGetData, "CheckSPVersion()");
    }

    // Release row(s)
    hr = pRowset->Release();

    pICommandText->Release();

    // Check the retrieved version
    if (strcmp(db_sp_version,sVersion))
        throw

new CTPCC_OLEDB_ERR(
    CTPCC_OLEDB_ERR::ERR_WRONG_SP_VERSION
);
}

void CTPCC_OLEDB::ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction,
LPCTSTR szLocation)
{
    HRESULT          hr;

    //char
    szState[6];
    char
    szMsg[SQL_MAX_MESSAGE_LEN];

H];

```

```

        char
        szTmp[6*SQL_MAX_MESSAGE_LEN
        COLEDBERR
        *pOLEDBErr;
        // not allocated until
needed (maybe never)
        int
        iLen;
        // Interfaces
        IErrorInfo*      pIErrorInfoAll
= NULL;
        IErrorInfo*
pIErrorInfoRecord = NULL;
        IErrorRecords*
pIErrorRecords = NULL;
        ISupportErrorInfo*
pISupportErrorInfo = NULL;
        ISQLServerErrorInfo*
pISQLServerErrorInfo = NULL;
        ISQLErrorInfo*
pISQLErrorInfo =
        NULL;

        // Information used when cannot
get custom error object
        ERRORINFO
        BasicErrorInfo;
        BSTR
        bstrDescription;
        // Number of error records.
        ULONG          nRecs;
        ULONG          nRec;

        // SQL Server error information
from ISQLServerErrorInfo.
        SSERRORINFO*
pSSErrorInfo = NULL;
        OLECHAR*
pSSErrorStrings = NULL;

        assert(pObjectWithError != NULL);

        pOLEDBErr = new
        COLEDBERR(szLocation);

        pOLEDBErr->m_NativeError = 0;
        pOLEDBErr->m_eAction = eAction;
        pOLEDBErr->m_bDeadLock =
        FALSE;

        szTmp[0] = 0;

        // Only ask for error information if
the interface supports it.
        // Note: SQLOLEDB provider
supports error interface, so this check is
// for good style only.
        hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**)
&pISupportErrorInfo);
        if (FAILED(hr))
        {
            _snprintf(szMsg,
            sizeof(szMsg), "SupportErrorInfo interface not
            supported (hr=0x%X)", hr);
            pOLEDBErr-
            >m_OLEDBErrStr = new char[strlen(szMsg)+1];
            strcpy(pOLEDBErr-
            >m_OLEDBErrStr, szMsg);
            throw pOLEDBErr;
        }

```

```

        /*if (FAILED(pISupportErrorInfo-
        >InterfaceSupportsErrorInfo(IID_InterfaceWithEr
        ror))
        {
            _snprintf(szMsg,
            sizeof(szMsg), "InterfaceWithError interface not
            supported");
            pOLEDBErr-
            >m_OLEDBErrStr = new char[strlen(szMsg)+1];
            strcpy(pOLEDBErr-
            >m_OLEDBErrStr, szMsg);
            return;
        }*/

        // Do not test the return of
GetErrorInfo. It can succeed and return
        // a NULL pointer in pIErrorInfoAll.
        Simply test the pointer.
        GetErrorInfo(0, &pIErrorInfoAll);

        if (pIErrorInfoAll != NULL)
        {
            // Test to see if it's a
            valid OLE DB IErrorInfo interface
            // exposing a list of
            records.
            if
            (SUCCEEDED(pIErrorInfoAll-
            >QueryInterface(IID_IErrorRecords, (void**)
            &pIErrorRecords)))
            {
                pIErrorRecords-
                >GetRecordCount(&nRecs);

                // Within
                each record, retrieve information from each
                // of the
                defined interfaces.
                for
                (nRec = 0; nRec < nRecs; nRec++)
                {
                    // Request the generic SQL error
                    interface.

                    pIErrorRecords-
                    >GetCustomErrorObject(nRec,

                    IID_ISQLErrorInfo, //
                    generic SQL error interface
                    (IUnknown**)
                    &pISQLErrorInfo);

                    if (pISQLErrorInfo != NULL)
                    {
                        // Request SQL
                        Server-specific error interface, not the generic
                        SQL error interface.

                        pISQLErrorInfo-
                        >QueryInterface(

                        IID_ISQLServerErrorInfo, // SQL
                        Server error interface
                        (void**)
                        &pISQLServerErrorInfo);

```

```

    }

    // Test to ensure the reference is
    valid, then

    // get error information from
    ISQLServerErrorInfo.

    if (pISQLServerErrorInfo != NULL)
    {
        pISQLServerErrorInfo-
        >GetErrorInfo(&pSSErrorInfo, &pSSErrorStrings);

        //
        ISQLServerErrorInfo::GetErrorInfo succeeds

        // even when it has
        nothing to return. Test the

        // pointers before
        using.

        if (pSSErrorInfo)
        {
            // First,
            add the error message.

            //
            Convert Unicode error string to ANSI.

            WideCharToMultiByte(CP_THREAD_
            ACP, 0,

            pSSErrorInfo->pwszMessage, -1,

            szMsg, sizeof(szMsg),

            NULL, NULL);

            // quit if
            there isn't enough room to concatenate error text

            if (
            (strlen(szMsg) + 2) > (sizeof(szTmp) -
            strlen(szTmp)) )

            break;

            //
            include line break after first error msg

            if
            (szTmp[0] != 0)

            strcat( szTmp, "\r\n");

            //
            concatenate the error record to the overall error
            message

```

```

        strcat(
        szTmp, szMsg );

        //
        Second, add the stored procedure name and line
        number, if available.

        if
        (wcslen(pSSErrorInfo->pwszProcedure)>0)
        {
            // Prefix with a line break

            iLen = sprintf(szMsg,
            "\r\nProcedure: ");

            // Convert Unicode error string to
            ANSI.

            WideCharToMultiByte(CP_THREAD_
            ACP, 0,

            pSSErrorInfo-
            >pwszProcedure, -1,

            &szMsg[iLen],
            sizeof(szMsg) - iLen,

            NULL, NULL);

            // Check if have space to add the
            line number.

            // Assume the line number takes no
            more than 3 digits.

            if ((strlen(szMsg) + 4)<
            sizeof(szMsg))

            {
                _snprintf(&szMsg[strlen(szMsg)],
                sizeof(szMsg),

                "%d",
                pSSErrorInfo->wLineNumber);

            }

            // quit if there isn't enough room to
            concatenate error text

```

```

        if ( ( strlen(szMsg) + 2) >
        (sizeof(szTmp) - strlen(szTmp)) )

        break;

        // concatenate the error record to
        the overall error message

        strcat( szTmp, szMsg );

        // copy the overall error string to
        the exception

        pOLEDBErr->m_OLEDBErrStr =
        new char[strlen(szTmp)+1];

        strcpy(pOLEDBErr-
        >m_OLEDBErrStr, szTmp);

        // Third,
        capture the (first) database error

        if
        (pOLEDBErr->m_NativeError == 0 &&
        pSSErrorInfo->INative != 0)

        {
            pOLEDBErr->m_NativeError =
            pSSErrorInfo->INative;

            // Check for deadlock error code
            and set the deadlock flag

            if (pSSErrorInfo->INative == 1205)

            {
                pOLEDBErr-
                >m_bDeadLock = TRUE;

            }

        }

        //
        IMalloc::Free needed to release references

        // on
        returned values.

        if
        (m_pIMalloc != NULL)

```

```

        {
            m_pIMalloc->Free(pSSErrStrings);

            m_pIMalloc->Free(pSSErrInfo);
        }
    }

    pISQLServerErrorInfo->Release();
}
else
{
    // Custom error object
    // Use general OLE-DB error interface.
    // Get the numeric error code
    pIErrorRecords->GetBasicErrorInfo(nRec, &BasicErrorInfo);

    if (pOLEDBErr->m_NativeError == 0)
    {
        // Get the failed call HRESULT code, which is not really the native error
        pOLEDBErr->m_NativeError = BasicErrorInfo.hrError;
    }

    // Try to get the string description of the error.
    pIErrorRecords->GetErrorInfo(nRec, LOCALE_USER_DEFAULT, (IErrorInfo*)&pIErrorInfoRecord);

    if (pIErrorInfoRecord)
    {
        pIErrorInfoRecord->GetDescription(&bstrDescription);

        //
        Convert Unicode error string to ANSI.
    }
}

```

```

        WideCharToMultiByte(CP_THREAD_
ACP, 0,
        bstrDescription, -1,
        szMsg, sizeof(szMsg),
        NULL, NULL);

    pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szMsg)+1];

    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}
// for()
} // if
(SUCCEEDED(pIErrorInfoAll->QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords)))
else
{
    // No
    IErrorRecords interface supported. Use default
    IErrorInfo.
    // Note:
    SQLOLEDB supports IErrorRecords, so this check
    is for good style only.
    _snprintf(szMsg, sizeof(szMsg),
"IErrorRecords interface not supported");

    pOLEDBErr->m_OLEDBErrStr =
new char[strlen(szMsg)+1];

    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}

pIErrorInfoAll->Release();
} // if (pIErrorInfoAll !=
NULL)
else
{
    // No IErrorInfo
    interface supported.
    // Note: SQLOLEDB
    supports IErrorInfo, so this check is for good
    style only.
    _snprintf(szMsg,
sizeof(szMsg), "IErrorInfo interface not
supported");
    pOLEDBErr->m_OLEDBErrStr = new char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}

    throw pOLEDBErr;
}

```

```

/*
 *
 * Create a new command object from
the SQL text passed in.
 *
 */
void CTPCC_OLEDB::CreateCommand(wchar_t*
szSQLCommand,
    // I: SQL query for the command

    ICommandText**
ppICommandText // O: returned
command object

)
{
    HRESULT
    hr;

    // Create a new command object
    hr = m_pIDBCreateCommand->CreateCommand(NULL, IID_ICommandText,
(IUnknown**)ppICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateComman
d, COLEDBERR::eCreateCommand,
"CTPCC_OLEDB::CreateCommand");
    }

    // Set command text
    hr = (*ppICommandText)->SetCommandText(DBGUID_SQL,
szSQLCommand);
    if (FAILED(hr))
    {
        ThrowError(*ppICommandText,
COLEDBERR::eSetCommandText,
"CTPCC_OLEDB::CreateCommand");
    }

    // Prepare the command
    PrepareCommand(*ppICommandText);
}

/*
 *
 * QueryInterface and Prepare in one
function for simplicity.
 * DEFERRED PREPARE property is set
to off to prepare immediately.
 */
void
CTPCC_OLEDB::PrepareCommand(ICommandText*
pICommandText)
{
    HRESULT
    hr;
    ICommandPrepare*
pICommandPrepare;
    ICommandProperties*
pICommandProperties;
    DBPROPSET
rowSetPropSet;
    DBPROP
rowSetProp;

    // Set the deferred prepare
property to false.
    rowSetProp.dwPropertyID =
SSPROP_DEFERPREPARE;
}

```

```

        memset(&rowSetProp.vValue, 0,
sizeof(rowSetProp.vValue));
        rowSetProp.dwOptions =
DBPROPOPTIONS_REQUIRED;
        rowSetProp.colid = DB_NULLID;

        rowSetPropSet.cProperties = 1;
        rowSetPropSet.guidPropertySet =
DBPROPSET_SQLSERVERROWSET;
        rowSetPropSet.rgProperties =
&rowSetProp;

// Query interface for setting
properties
        hr = piCommandText-
>QueryInterface(IID_ICommandProperties, (void
***)&piCommandProperties);
        if (FAILED(hr))
        {
                ThrowError(piCommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
        }

// Set the property set
        hr = piCommandProperties-
>SetProperties(1, &rowSetPropSet);
        if (FAILED(hr))
        {
                ThrowError(piCommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
        }

// Get interface for preparing
commands
        hr = piCommandText-
>QueryInterface(IID_ICommandPrepare, (void
***)&piCommandPrepare);
        if (FAILED(hr))
        {
                ThrowError(piCommandText,
COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
        }

// Prepare Payment command
        hr = piCommandPrepare-
>Prepare(0xFFFFFFFF);
        if (FAILED(hr))
        {
                ThrowError(piCommandPrepare,
COLEDBERR::ePrepare,
"CTPCC_OLEDB::PrepareCommand");
        }
}

/*
 * Initialize fields of an array of
bindings structures.
 * Needs to be called before setting
individual parameter/column bindings.
 */
void CTPCC_OLEDB::InitBindings(DBBINDING*
pDBBindings, // IO: array of bindings

        int iCount, // I:
number of elements in the array

```

```

        eBindingType BindingType)
// I: what the bindings will be used
for (parameters/columns)
int i;

        for(i = 0; i < iCount; i++)
        {
                pDBBindings[i].iOrdinal = i + 1;
                pDBBindings[i].obLength = 0;
                pDBBindings[i].obStatus = 0;
                pDBBindings[i].pTypeInfo = NULL;
                pDBBindings[i].pObject = NULL;
                pDBBindings[i].pBindExt = NULL;
                pDBBindings[i].dwPart = DBPART_VALUE;

                switch (BindingType)
                {
                        case
eInputParameter:
                        pDBBindings[i].eParamIO =
DBPARAMIO_INPUT;
                        break;
                        case
eOutputParameter:
                        pDBBindings[i].eParamIO =
DBPARAMIO_OUTPUT;
                        break;
                        case
eInputOutputParameter:
                        pDBBindings[i].eParamIO =
DBPARAMIO_INPUT | DBPARAMIO_OUTPUT;
                        break;
                        case eOutputColumn:
                        pDBBindings[i].eParamIO =
DBPARAMIO_NOTPARAM;
                        break;
                        default:
                        assert(false); // this should never
happen
                }

                pDBBindings[i].dwMemOwner =
DBMEMOWNER_CLIENTOWNED;
                pDBBindings[i].dwFlags = 0;

                pDBBindings[i].bPrecision = 0;
                pDBBindings[i].bScale = 0;
        }
}

/*
 * Perform binding for one parameter
or output column.
 */
void CTPCC_OLEDB::SetBinding(DBBINDING*
pDBBinding, // I: binding row structure

        size_t obValue, // I: parameter
(column) offset in the user buffer

        size_t cbMaxLen, // I: parameter (column) length

```

```

        DBTYPE wType // I: parameter
(column) type
)
{
        pDBBinding->obValue =
(ULONG)obValue;
        pDBBinding->cbMaxLen =
(ULONG)cbMaxLen;
        pDBBinding->wType = wType;
}

void CTPCC_OLEDB::InitStockLevelParams()
{
        int
i;
        HRESULT
hr;
        wchar_t
szName[IMAX_SP_NAME_LEN];
        IAccessor*
piAccessor;

        const ULONG
nInputParams = 3;
// input parameters
        const ULONG
nOutputParams = 1;
// output 1st result set columns

// Structure to bind in accessor
DBBINDING
acInputDBBinding[nInputParams];
DBBINDSTATUS
acInputDBBindStatus[nInputParams];
DBBINDING
acOutputDBBinding[nOutputParams];
DBBINDSTATUS
acOutputDBBindStatus[nOutputPara
ms];

// Set command text
_snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call
%stpcck_stocklevel(?,?,?)", m_szSPPrefix);

// Create and Prepare a new
command object for StockLevel.
CreateCommand(szName,
&m_piStockLevelCommand);

// Describe the consumer buffer by
filling in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

        i = 0;
// StockLevel parameter 1

```

```

        SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, w_id),
sizeof(m_txn.StockLevel.w_id), DBTYPE_I4);

// StockLevel parameter 2
        SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, d_id),
sizeof(m_txn.StockLevel.d_id), DBTYPE_UI1);

// StockLevel parameter 3
        SetBinding(&acInputDBBinding[i++
], offsetof(STOCK_LEVEL_DATA, threshold),
sizeof(m_txn.StockLevel.threshold), DBTYPE_I2);

        hr = m_pIStockLevelCommand-
>QueryInterface(IID_IAccessor, (void
**) &pIAccessor);
        if (FAILED(hr))
        {
                ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eQueryInterface,
"InitStockLevelParams()");
        }

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
        nInputParams,
        acInputDBBinding,
        sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelInputAccessor,
        acInputDBBindStatus);
        if (FAILED(hr))
        {
                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
        }

        m_StockLevelExecuteParams.cPara
mSets = 1;
        m_StockLevelExecuteParams.hAcc
essor = m_hStockLevelInputAccessor;
        m_StockLevelExecuteParams.pData
= &m_txn.StockLevel;

// Now fill the binding information
for result set 1 output columns
        InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

// Binding for a rowset that may
return more than one row.
        i = 0;
// StockLevel output column 1
        SetBinding(&acOutputDBBinding[i+
+], offsetof(STOCK_LEVEL_DATA, low_stock),
sizeof(m_txn.StockLevel.low_stock),
DBTYPE_I4);

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
        nOutputParams,
        acOutputDBBinding,
        sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelOutputAccessor,
        acOutputDBBindStatus);
        if (FAILED(hr))
        {

```

```

                ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
        }

void CTPCC_OLEDB::StockLevel()
{
        HRESULT
        hr;
        int
                iTryCount = 0;

        IRowset*
        pRowset;
        LONG
        cRows = 1; // number of rows
returned in the rowset
        ULONG
        cRowsObtained;
        HROW
        rghRow;
//returned row handles
        HROW*
        prghRow = &rghRow;

        while (TRUE)
        {
                try
                {
                        //
Execute the prepared command
                        hr =
m_pIStockLevelCommand->Execute(NULL,
IID_IRowset, &m_StockLevelExecuteParams,
NULL,

                (IUnknown
                **)&pRowset);
                        if
(FAILED(hr))
                        {
                                ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eExecute, "StockLevel()");
                        }

// Fetch
the result row handle(s)
                        hr =
pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
                        if
(FAILED(hr))
                        {
                                ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eGetNextRows, "StockLevel()");
                        }

// Fetch
the actual row data by handle
                        hr =
pRowset->GetData(rghRow,
m_hStockLevelOutputAccessor,
&m_txn.StockLevel);
                        if
(FAILED(hr))
                        {
                                ThrowError(m_pIStockLevelComma
nd, COLEDBERR::eGetData, "StockLevel()");
                        }

```

```

//
                Release row(s)
                hr =
pRowset->ReleaseRows(cRowsObtained,
prghRow, NULL, NULL, NULL);
//
                Release rowset
                hr =
pRowset->Release();

        m_txn.StockLevel.exec_status_code
= eOK;
                break;
        }
        catch (COLEDBERR
*e)
        {
                if ((!e-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))
                throw;

// hit
                deadlock; backoff for increasingly longer period
                delete e;
                Sleep(10
                * iTryCount);
        }

//
        if (iTryCount)
        throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{
        int
                i, j, iOICount;
        HRESULT
                hr;
        wchar_t
                szName[iMAX_SP_NAME_LEN];
        IAccessor*
                pIAccessor;

        const ULONG
                nInputParams = 5 +
3*MAX_OL_NEW_ORDER_ITEMS; // input
parameters
        const ULONG
                nOutputParams = 5;
// output 1st result set columns
        const ULONG
                nOutputParams2 = 8;
// output 2nd result set columns
// Structure to bind in accessor
        DBBINDING
                acInputDBBinding[nInputParams];
        DBBINDSTATUS
                acInputDBBindStatus[nInputParams];
        DBBINDING
                acOutputDBBinding[nOutputParams];
}

```

```

DBBINDSTATUS
acOutputDBBindStatus[nOutputPara
ms];
DBBINDING
acOutputDBBinding2[nOutputParam
s2];
DBBINDSTATUS
acOutputDBBindStatus2[nOutputPar
ams2];

// Describe the consumer buffer by
filling in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

i = 0;
// NewOrder parameter 1
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, w_id),
sizeof(m_txn.NewOrder.w_id), DBTYPE_I4);

// NewOrder parameter 2
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, d_id),
sizeof(m_txn.NewOrder.d_id), DBTYPE_UI1);

// NewOrder parameter 3
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, c_id),
sizeof(m_txn.NewOrder.c_id), DBTYPE_I4);

// NewOrder parameter 4
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

// NewOrder parameter 5
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, o_all_local),
sizeof(m_txn.NewOrder.o_all_local),
DBTYPE_UI1);

for (j=0;
j<MAX_OL_NEW_ORDER_ITEMS; j++)
{
SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA, OL[j].ol_i_id),
sizeof(m_txn.NewOrder.OL[j].ol_i_id),
DBTYPE_I4);

SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA,
OL[j].ol_supply_w_id),
sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id),
DBTYPE_I4);

SetBinding(&acInputDBBinding[i++
], offsetof(NEW_ORDER_DATA,
OL[j].ol_quantity),
sizeof(m_txn.NewOrder.OL[j].ol_quantity),
DBTYPE_I2);
}

// Now fill the binding information
for result set 1 output columns
InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

```

```

// Binding for the order line rowsets
(each consist of one row).
// Bind to offsets of the
OL_NEW_ORDER_DATA structure instead of
NEW_ORDER_DATA.
// IRowset::GetData() will be
passed individual array slots OL[i] to fetch the
data
// from the row set.

i = 0;
// NewOrder output column 1
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name),
DBTYPE_STR);

// NewOrder output column 2
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock),
DBTYPE_I2);

// NewOrder output column 3
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA,
ol_brand_generic),
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic),
DBTYPE_STR);

// NewOrder output column 4
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA, ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price),
DBTYPE_R8);

// NewOrder output column 5
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_NEW_ORDER_DATA,
ol_amount),
sizeof(m_txn.NewOrder.OL[0].ol_amount),
DBTYPE_R8);

// Now fill the binding information
for result set 2 output columns
InitBindings(&acOutputDBBinding2[
0], nOutputParams2, eOutputColumn);

i = 0;
// NewOrder output column 1
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, w_tax),
sizeof(m_txn.NewOrder.w_tax), DBTYPE_R8);

// NewOrder output column 2
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, d_tax),
sizeof(m_txn.NewOrder.d_tax), DBTYPE_R8);

// NewOrder output column 3
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, o_id),
sizeof(m_txn.NewOrder.o_id), DBTYPE_I4);

// NewOrder output column 4
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_last),
sizeof(m_txn.NewOrder.c_last), DBTYPE_STR);

// NewOrder output column 5
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_discount),
sizeof(m_txn.NewOrder.c_discount),
DBTYPE_R8);

```

```

// NewOrder output column 6
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, c_credit),
sizeof(m_txn.NewOrder.c_credit), DBTYPE_STR);

// NewOrder output column 7
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA, o_entry_d),
sizeof(m_txn.NewOrder.o_entry_d),
DBTYPE_DBTIMESTAMP);

// NewOrder output column 8
SetBinding(&acOutputDBBinding2[i
++], offsetof(NEW_ORDER_DATA,
o_commit_flag),
sizeof(m_txn.NewOrder.o_commit_flag),
DBTYPE_I2);

for (j=0;
j<MAX_OL_NEW_ORDER_ITEMS; j++)
{
// Set command text
first

// Print the fixed first
portion of parameters
i =
_snowprintf(szName,
sizeof(szName)/sizeof(szName[0]),

L"{call
%stppc_neworder (?,?,?,?," m_szSPPrefix);

// Now print the
variable portion depending on the number of
order line parameters
for (iOLCount = 0;
iOLCount <= j; ++iOLCount)
{
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",?,?,"");
}

// Print the fixed end
if (j !=
MAX_OL_NEW_ORDER_ITEMS - 1)
{
//
append 'default' for the parameters that are not
used
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i,
L",default");
}
else // using
all 15 order line parameters
{
i +=
_snowprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",");
}

// Create and Prepare
a new command object for NewOrder.

CreateCommand(szName,
&m_pINewOrderCommand[j]);

// Now create the
input accessor for this prepared command

```



```

        hr =
m_pINewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void
**) &pIAccessor);
        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderComma
nd[j], COLEDBERR::eQueryInterface,
"InitNewOrderParams()");
        }
        hr = pIAccessor-
>CreateAccessor(
        DBACCESSOR_PARAMETERDATA,
        5 + 3 * (j + 1),
        acInputDBBinding,
        sizeof(NEW_ORDER_DATA),
        &m_hNewOrderInputAccessor[j],
        acInputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
        }
        m_NewOrderExecuteParams[j].cPar
amSets = 1;
        //
m_NewOrderExecuteParams.hAccessor is set
dynamically at run-time
        // based on the
number of new order items for the particular
transaction call.
        m_NewOrderExecuteParams[j].hAcc
essor = m_hNewOrderInputAccessor[j];
        m_NewOrderExecuteParams[j].pDat
a = &m_txn.NewOrder;
        // Create accessor for
the first rowset
        hr = pIAccessor-
>CreateAccessor(
        DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,

```

```

        sizeof(OL_NEW_ORDER_DATA),
&m_hNewOrderOutputAccessor[j],
        acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
        }
        // Create accessor for
the second rowset
        hr = pIAccessor-
>CreateAccessor(
        DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
        nOutputParams2,
        acOutputDBBinding2,
        sizeof(NEW_ORDER_DATA),
        &m_hNewOrderOutputAccessor2[j],
        acOutputDBBindStatus2);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
        }
        pIAccessor-
>Release();
    }
}
void CTPCC_OLEDB::NewOrder()
{
    HRESULT hr;
    int
    iTryCount = 0;
    IMultipleResults*
pMultipleResults;
    IRowset*
pRowset;
    IRowset*
pRowset2;
    LONG
    cRows = 1; //
number of rows returned in the 1st rowset
    ULONG
    cRowsObtained;
    HROW
    rghRows;
    //returned row handles for the 1st
result set
    HROW*
    prghRows =
    &rghRows;
    LONG
    cRows2 = 1; //
number of rows returned in the 2nd rowset
    ULONG
    cRowsObtained2;

```

```

    HROW
    rghRows2;
    //returned row handle for the 2nd
result set
    HROW*
    prghRows2 =
    &rghRows2;
    int
    i;
    long
    lRowsAffected;
    // the number of affected rows for
a rowset
    int
    iHandleIndex; // index
into the handle arrays based on the orders count
        // check whether any order lines
are for a remote warehouse
        m_txn.NewOrder.o_all_local = 1;
        for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
        {
            if
(m_txn.NewOrder.OL[i].ol_supply_w_id !=
m_txn.NewOrder.w_id)
            {
                m_txn.NewOrder.o_all_local = 0;
                // at least one remote warehouse
                break;
            }
        }
        iHandleIndex =
m_txn.NewOrder.o_ol_cnt - 1; // for
convenience
        while (TRUE)
        {
            try
            {
                //
Execute the prepared command (according to the
number of new orders)
                // Ask
for IMultipleResults because it returns 2 rowsets.
                hr =
m_pINewOrderCommand[iHandleIndex]-
>Execute(
                NULL,
                IID_IMultipleResults,
                &m_NewOrderExecuteParams[iHan
dleIndex],
                NULL,
                (IUnknown
**) &pMultipleResults);
                if
(FAILED(hr))
            {

```

```

        ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eExecute,
"NewOrder()");
    }

    //////////////////////////////////////
    // Get
order line results

    //////////////////////////////////////

    m_txn.NewOrder.total_amount = 0;
    for (i =
0; i < m_txn.NewOrder.o_ol_cnt; ++i)
    {

        // Get the first rowset object

        hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset,
&IRowsAffected, (IUnknown **)&pRowset);

        if (FAILED(hr))
        {

            char szTmp[256];

            _snprintf(szTmp,
sizeof(szTmp), "NewOrder() result set %d,
hr=0x%X", i, hr);

            ThrowError(m_pINewOrderComma
nd[m_txn.NewOrder.o_ol_cnt - 1],
COLEDBERR::eGetResult, szTmp);

        }

        // Fetch the result row handle(s)

        hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);

        if (FAILED(hr))
        {

            ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eGetNextRows,
"NewOrder()");

        }

        // Fetch the actual row data by
handle

        hr = pRowset->GetData(rghRows,
m_hNewOrderOutputAccessor[iHandleIndex],
&m_txn.NewOrder.OL[i]);

        if (FAILED(hr))
        {

```

```

        ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eGetData,
"NewOrder()");
    }

    m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;

    // Release row(s)

    hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL,
NULL, NULL);

    // Release rowset

    hr = pRowset->Release();

    }

    //////////////////////////////////////
    // Get
the second rowset object

    //////////////////////////////////////

    hr =
pMultipleResults->GetResult(NULL, 0,
IID_IRowset, &IRowsAffected, (IUnknown
**)&pRowset2);

    if
(FAILED(hr))
    {

        char szTmp[256];

        _snprintf(szTmp, sizeof(szTmp),
"NewOrder() result set %d, hr=%d", i, hr);

        ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eGetResult,
szTmp);

    }

    // Fetch
the result row handle(s)

    hr =
pRowset2->GetNextRows(DB_NULL_HCHAPTER,
0, cRows2, &cRowsObtained2, &prghRows2);

    if
(FAILED(hr))
    {

        ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eGetNextRows,
"NewOrder()");

    }

    // Fetch
the actual row data by handle

    hr =
pRowset2->GetData(rghRows2,
m_hNewOrderOutputAccessor2[iHandleIndex],
&m_txn.NewOrder);

    if
(FAILED(hr))
    {

```

```

        ThrowError(m_pINewOrderComma
nd[iHandleIndex], COLEDBERR::eGetData,
"NewOrder()");
    }

    //
Release row(s)

    hr =
pRowset2->ReleaseRows(cRowsObtained2,
prghRows2, NULL, NULL, NULL);

    //
Release rowset

    hr =
pRowset2->Release();

    //
Release the common MultipleResults interface

    hr =
pMultipleResults->Release();

    if
(m_txn.NewOrder.o_all_local == 1)
    {

        m_txn.NewOrder.total_amount *=
((1 + m_txn.NewOrder.w_tax +
m_txn.NewOrder.d_tax) * (1 -
m_txn.NewOrder.c_discount));

        m_txn.NewOrder.exec_status_code
= eOK;

    }
    else
    {

        m_txn.NewOrder.exec_status_code
= eInvalidItem;

    }

    break;

}
catch (COLEDBERR
*e)
{

    if ((!e-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))

        throw;

    // hit
deadlock; backoff for increasingly longer period
delete e;
Sleep(10

    * iTryCount);

}

//
if (iTryCount)
//
throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitPaymentParams()
{

    int
i;
HRESULT
hr;

    wchar_t
szName[iMAX_SP_NAME_LEN];

```

```

IAccessor*
    pIAccessor;

const ULONG
    nInputParams = 7;
// input parameters
const ULONG
    nOutputParams = 27;
// output result set columns
// Structure to bind in accessor
DBBINDING
    acInputDBBinding[nInputParams];
DBBINDSTATUS
    acInputDBBindStatus[nInputParams
];
DBBINDING
    acOutputDBBinding[nOutputParams
];
DBBINDSTATUS
    acOutputDBBindStatus[nOutputPara
ms];

// Set command text
_snowprintf(szName,
sizeof(szName)/sizeof(szName[0]), L"{call
%stpc_payment(?,?,?,?,?)", m_szSPPrefix);

// Create and Prepare a new
command object for Payment.
CreateCommand(szName,
&m_pIPaymentCommand);

// Describe the consumer buffer by
filling in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

i = 0;
// Payment parameter 1
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, w_id),
sizeof(m_txn.Payment.w_id), DBTYPE_I4);

// Payment parameter 2
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, c_w_id),
sizeof(m_txn.Payment.c_w_id), DBTYPE_I4);

// Payment parameter 3
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, h_amount),
sizeof(m_txn.Payment.h_amount), DBTYPE_R8);

// Payment parameter 4
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, d_id),
sizeof(m_txn.Payment.d_id), DBTYPE_UI1);

// Payment parameter 5
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, c_d_id),
sizeof(m_txn.Payment.c_d_id), DBTYPE_UI1);

// Payment parameter 6
SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

```

```

SetBinding(&acInputDBBinding[i++
], offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

hr = m_pIPaymentCommand-
>QueryInterface(IID_IAccessor, (void
**) &pIAccessor);
if (FAILED(hr))
{
    ThrowError(m_pIPaymentCommand
, COLEDBERR::eQueryInterface,
"InitPaymentParams()");
}

hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(PAYMENT_DATA),
&m_hPaymentInputAccessor,
acInputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitPaymentParams()");
}

m_PaymentExecuteParams.cParam
Sets = 1;
m_PaymentExecuteParams.hAccess
or = m_hPaymentInputAccessor;
m_PaymentExecuteParams.pData =
&m_txn.Payment;

// Now fill the binding information
for output columns
InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

i = 0;
// Payment output column 1

SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

// Payment output column 2
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

// Payment output column 3
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, h_date),
sizeof(m_txn.Payment.h_date),
DBTYPE_DBTIMESTAMP);

// Payment output column 4
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_street_1),
sizeof(m_txn.Payment.w_street_1),
DBTYPE_STR);

// Payment output column 5
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_street_2),
sizeof(m_txn.Payment.w_street_2),
DBTYPE_STR);

// Payment output column 6

```

```

SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_city),
sizeof(m_txn.Payment.w_city), DBTYPE_STR);
// Payment output column 7
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_state),
sizeof(m_txn.Payment.w_state), DBTYPE_STR);

// Payment output column 8
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, w_zip),
sizeof(m_txn.Payment.w_zip), DBTYPE_STR);

// Payment output column 9
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1),
DBTYPE_STR);

// Payment output column 10
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2),
DBTYPE_STR);

// Payment output column 11
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 12
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 13
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 14
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_first),
sizeof(m_txn.Payment.c_first), DBTYPE_STR);

// Payment output column 15
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_middle),
sizeof(m_txn.Payment.c_middle), DBTYPE_STR);

// Payment output column 16
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1),
DBTYPE_STR);

// Payment output column 17
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2),
DBTYPE_STR);

// Payment output column 18
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 19
SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 20

```

```

        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

        // Payment output column 21
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_phone),
sizeof(m_txn.Payment.c_phone), DBTYPE_STR);

        // Payment output column 22
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_since),
sizeof(m_txn.Payment.c_since),
DBTYPE_DBTIMESTAMP);

        // Payment output column 23
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_credit),
sizeof(m_txn.Payment.c_credit), DBTYPE_STR);

        // Payment output column 24
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_credit_lim),
sizeof(m_txn.Payment.c_credit_lim),
DBTYPE_R8);

        // Payment output column 25
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_discount),
sizeof(m_txn.Payment.c_discount), DBTYPE_R8);

        // Payment output column 26
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_balance),
sizeof(m_txn.Payment.c_balance), DBTYPE_R8);

        // Payment output column 27
        SetBinding(&acOutputDBBinding[i+
+], offsetof(PAYMENT_DATA, c_data),
sizeof(m_txn.Payment.c_data), DBTYPE_STR);

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(PAYMENT_DATA),

&m_hPaymentOutputAccessor,
acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitPaymentParams()");
        }
    }

void CTPCC_OLEDB::Payment()
{
    HRESULT
    hr;
    int
        iTryCount = 0;

    IRowset*
    pRowset;
    LONG
    cRows = 1; // number of rows
    returned in the rowset
    ULONG
    cRowsObtained;
    HROW
    rghRow;
    //returned row handles

```

```

    HROW*
    prghRow = &rghRow;

    if (m_txn.Payment.c_id != 0)

        m_txn.Payment.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            //
            Execute the prepared command
            hr =
            m_pIPaymentCommand->Execute(NULL,
IID_IRowset, &m_PaymentExecuteParams,
NULL,

                (IUnknown
**)&pRowset);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIPaymentCommand
, COLEDBERR::eExecute, "Payment()");
            }

            // Fetch
            the result row handle(s)
            hr =
            pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIPaymentCommand
, COLEDBERR::eGetNextRows, "Payment()");
            }

            // Fetch
            the actual row data by handle
            hr =
            pRowset->GetData(rghRow,
m_hPaymentOutputAccessor, &m_txn.Payment);
            if
            (FAILED(hr))
            {
                ThrowError(m_pIPaymentCommand
, COLEDBERR::eGetData, "Payment()");
            }

            //
            Release row(s)
            hr =
            pRowset->ReleaseRows(cRowsObtained,
prghRow, NULL, NULL);
            //
            Release rowset
            hr =
            pRowset->Release();

            if
            (m_txn.Payment.c_id == 0)

                throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
            else

                m_txn.Payment.exec_status_code
= eOK;

```

```

                break;
            }
            catch (COLEDBERR
*e)
            {
                if ((le-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))

                    throw;

                // hit
                deadlock; backoff for increasingly longer period
                delete e;
                Sleep(10
*iTryCount);
            }

            //
            if (iTryCount)
            //
            throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
        }

        void CTPCC_OLEDB::InitOrderStatusParams()
        {
            int
            i;
            HRESULT
            hr;
            wchar_t
            szName[iMAX_SP_NAME_LEN];
            IAccessor*
            pIAccessor;

            const ULONG
            nInputParams = 4;
            // input parameters
            const ULONG
            nOutputParams = 5;
            // output 1st result set columns
            const ULONG
            nOutputParams2 = 8;
            // output 2nd result set columns
            // Structure to bind in accessor
            DBBINDING
            acInputDBBinding[nInputParams];
            DBBINDSTATUS
            acInputDBBindStatus[nInputParams
];
            DBBINDING
            acOutputDBBinding[nOutputParams
];
            DBBINDSTATUS
            acOutputDBBindStatus[nOutputPara
ms];
            DBBINDING
            acOutputDBBinding2[nOutputParam
s2];
            DBBINDSTATUS
            acOutputDBBindStatus2[nOutputPar
ams2];

            // Set command text
            _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),

```

```

L"(call
%stpc_orderstatus(?,?,?,?), m_szSPPrefix);

// Create and Prepare a new
command object for OrderStatus.
CreateCommand(szName,
&m_pIOrderStatusCommand);

// Describe the consumer buffer by
filling in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

i = 0;
// OrderStatus parameter 1
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, w_id),
sizeof(m_txn.OrderStatus.w_id), DBTYPE_I4);

// OrderStatus parameter 2
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, d_id),
sizeof(m_txn.OrderStatus.d_id), DBTYPE_UI1);

// OrderStatus parameter 3
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus parameter 4
SetBinding(&acInputDBBinding[i++
], offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

hr = m_pIOrderStatusCommand-
>QueryInterface(IID_IAccessor, (void
**) &pIAccessor);
if (FAILED(hr))
{
ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eQueryInterface,
"InitOrderStatusParams()");
}

hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(ORDER_STATUS_DATA),
&m_hOrderStatusInputAccessor,
acInputDBBindStatus);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

m_OrderStatusExecuteParams.cPar
amSets = 1;
m_OrderStatusExecuteParams.hAcc
essor = m_hOrderStatusInputAccessor;
m_OrderStatusExecuteParams.pDat
a = &m_txn.OrderStatus;

```

```

// Now fill the binding information
for result set 1 output columns
InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

// Binding for a rowset that may
return more than one row.
// Bind to offsets of the
OL_ORDER_STATUS_DATA structure instead of
ORDER_STATUS_DATA.
// IRowset::GetData() will be
passed individual array slots OL[i] to fetch the
data
// from the row set.

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_supply_w_id),
sizeof(m_txn.OrderStatus.OL[0].ol_supply_w_id),
DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_txn.OrderStatus.OL[0].ol_i_id),
DBTYPE_I4);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_quantity),
sizeof(m_txn.OrderStatus.OL[0].ol_quantity),
DBTYPE_I2);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_amount),
sizeof(m_txn.OrderStatus.OL[0].ol_amount),
DBTYPE_R8);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding[i+
+], offsetof(OL_ORDER_STATUS_DATA,
ol_delivery_d),
sizeof(m_txn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_ORDER_STATUS_DATA),
&m_hOrderStatusOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

// Now fill the binding information
for result set 2 output columns
InitBindings(&acOutputDBBinding2[
0], nOutputParams2, eOutputColumn);

i = 0;
// OrderStatus output column 1

```

```

SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);
// OrderStatus output column 2
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_txn.OrderStatus.c_first), DBTYPE_STR);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_txn.OrderStatus.c_middle),
DBTYPE_STR);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
o_entry_d),
sizeof(m_txn.OrderStatus.o_entry_d),
DBTYPE_DBTIMESTAMP);

// OrderStatus output column 7
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
o_carrier_id),
sizeof(m_txn.OrderStatus.o_carrier_id),
DBTYPE_I2);

// OrderStatus output column 8
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA,
c_balance), sizeof(m_txn.OrderStatus.c_balance),
DBTYPE_R8);

// OrderStatus output column 9
SetBinding(&acOutputDBBinding2[i
++], offsetof(ORDER_STATUS_DATA, o_id),
sizeof(m_txn.OrderStatus.o_id), DBTYPE_I4);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),
&m_hOrderStatusOutputAccessor2,
acOutputDBBindStatus2);
if (FAILED(hr))
{
ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

void CTPCC_OLEDB::OrderStatus()
{
HRESULT
hr;

int
iTryCount = 0;
IMultipleResults*
pMultipleResults;
IRowset*
pRowset;
IRowset*
pRowset2;

```

```

LONG
        cRows =
MAX_OL_ORDER_STATUS_ITEMS; //
number of rows returned in the 1st rowset
ULONG
        cRowsObtained;
HROW
        rghRows[MAX_OL_ORDER_STATUS
_ITEMS]; //returned row
handles for the 1st result set
HROW*
        prghRows =
&rghRows[0];
LONG
        cRows2 = 1; //
number of rows returned in the 2nd rowset
ULONG
        cRowsObtained2;
HROW
        rghRows2;
//returned row handle for the 2nd
result set
HROW*
        prghRows2 =
&rghRows2;
int
        i;
long
        iRowsAffected;
// the number of affected rows for
a rowset
if (m_txn.OrderStatus.c_id != 0)
    m_txn.OrderStatus.c_last[0] = 0;
while (TRUE)
{
    try
    {
        //
        Execute the prepared command
        // Ask
        for IMultipleResults because it returns 2 rowsets.
        hr =
        m_pIOrderStatusCommand->Execute(NULL,
IID_IMultipleResults,
&m_OrderStatusExecuteParams, NULL,
        (IUnknown
        **)&pMultipleResults);
        if
        (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eExecute, "OrderStatus()");
        }
        // Get
        order line results
        // Get
        the first rowset object

```

```

        hr =
        pMultipleResults->GetResult(NULL, 0,
IID_IRowset, &iRowsAffected, (IUnknown
        **)&pRowset);
        if
        (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetResult, "OrderStatus()");
        }
        // Fetch
        the result row handle(s)
        hr =
        pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRows);
        if
        (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetNextRows,
"OrderStatus()");
        }
        m_txn.OrderStatus.o_ol_cnt =
(short)cRowsObtained;
        // Get
        the data from multiple rows in this rowset
        for (i =
0; i < m_txn.OrderStatus.o_ol_cnt; ++i)
        {
            // Fetch the actual row data by
            handle
            hr = pRowset-
>GetData(rghRows[i],
m_hOrderStatusOutputAccessor,
&m_txn.OrderStatus.OL[i]);
            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetData, "OrderStatus()");
            }
        }
        //
        Release row(s)
        hr =
        pRowset->ReleaseRows(cRowsObtained,
prghRows, NULL, NULL);
        //
        Release rowset
        hr =
        pRowset->Release();
        // Get
        the second rowset object
        if
        (m_txn.OrderStatus.o_ol_cnt > 0)
        {

```

```

        hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset,
&iRowsAffected, (IUnknown **)&pRowset2);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetResult, "OrderStatus()");
        }
        // Fetch the result row handle(s)
        hr = pRowset2-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetNextRows,
"OrderStatus()");
        }
        // Fetch the actual row data by
        handle
        hr = pRowset2-
>GetData(rghRows2,
m_hOrderStatusOutputAccessor2,
&m_txn.OrderStatus);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusComm
and, COLEDBERR::eGetData, "OrderStatus()");
        }
        // Release row(s)
        hr = pRowset2->Release();
        //
        Release the common MultipleResults interface
        hr =
        pMultipleResults->Release();
        if
        (m_txn.OrderStatus.o_ol_cnt == 0)
        {
            throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_NO_SUCH_ORDER );
        }
        else if
        (m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
        {
            throw new CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
        }
        else

```

```

        m_txn.OrderStatus.exec_status_cod
e = eOK;
                                break;
                                }
                                catch (COLEDBERR
*e)
                                {
                                if (!(!e-
>m_bDeadLock) || (++iTryCount >
iMaxRetries))
                                throw;

                                // hit
                                deadlock; backoff for increasingly longer period
                                delete e;
                                Sleep(10
*iTryCount);
                                }

                                //
                                // if (iTryCount)
                                // throw new
                                CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
ETRIED_TRANS, iTryCount);
                                }

                                void CTPCC_OLEDB::InitDeliveryParams()
                                {
                                int
                                i;
                                HRESULT
                                hr;
                                wchar_t
                                szName[iMAX_SP_NAME_LEN];
                                IAccessor*
                                pIAccessor;

                                const ULONG
                                nInputParams = 2;
                                // input parameters
                                const ULONG
                                nOutputParams = 10;
                                // output 1st result set columns

                                // Structure to bind in accessor
                                DBBINDING
                                acInputDBBinding[nInputParams];
                                DBBINDSTATUS
                                acInputDBBindStatus[nInputParams
];
                                DBBINDING
                                acOutputDBBinding[nOutputParams
];
                                DBBINDSTATUS
                                acOutputDBBindStatus[nOutputPara
ms];

                                // Set command text
                                _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call
%stpc_delivery (?;?)", m_szSPPrefix);

                                // Create and Prepare a new
                                command object for Delivery.

```

```

                                CreateCommand(szName,
                                &m_pIDeliveryCommand);
                                // Describe the consumer buffer by
                                filling in the array
                                // of DBBINDING structures. Each binding
                                associates
                                // a single parameter to the consumer's buffer.
                                InitBindings(&acInputDBBinding[0],
                                nInputParams, eInputParameter);

                                i = 0;
                                // Delivery parameter 1
                                SetBinding(&acInputDBBinding[i++
], offsetof(DELIVERY_DATA, w_id),
                                sizeof(m_txn.Delivery.w_id), DBTYPE_I4);

                                // Delivery parameter 2
                                SetBinding(&acInputDBBinding[i++
], offsetof(DELIVERY_DATA, o_carrier_id),
                                sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);

                                hr = m_pIDeliveryCommand-
>QueryInterface(IID_IAccessor, (void
**) &pIAccessor);
                                if (FAILED(hr))
                                {
                                ThrowError(m_pIDeliveryCommand,
                                COLEDBERR::eQueryInterface,
                                "InitDeliveryParams()");
                                }

                                hr = pIAccessor->CreateAccessor(
                                DBACCESSOR_PARAMETERDATA,
                                nInputParams,
                                acInputDBBinding,
                                sizeof(DELIVERY_DATA),
                                &m_hDeliveryInputAccessor,
                                acInputDBBindStatus);
                                if (FAILED(hr))
                                {
                                ThrowError(pIAccessor,
                                COLEDBERR::eCreateAccessor,
                                "InitDeliveryParams()");
                                }

                                m_DeliveryExecuteParams.cParamS
ets = 1;
                                m_DeliveryExecuteParams.hAccesso
r = m_hDeliveryInputAccessor;
                                m_DeliveryExecuteParams.pData =
                                &m_txn.Delivery;

                                // Now fill the binding information
                                for result set 1 output columns
                                InitBindings(&acOutputDBBinding[0
], nOutputParams, eOutputColumn);

                                // Binding for a rowset that may
                                return more than one row.
                                for (i = 0; i < 10; ++i)
                                {
                                // Delivery output
                                column 1

                                SetBinding(&acOutputDBBinding[i],
                                offsetof(DELIVERY_DATA, o_id[i]),
                                sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
                                }

                                hr = pIAccessor->CreateAccessor(
                                DBACCESSOR_ROWDATA |
                                DBACCESSOR_OPTIMIZED,

```

```

                                acOutputDBBinding,
                                sizeof(DELIVERY_DATA),
                                &m_hDeliveryOutputAccessor,
                                acOutputDBBindStatus);
                                if (FAILED(hr))
                                {
                                ThrowError(pIAccessor,
                                COLEDBERR::eCreateAccessor,
                                "InitDeliveryParams()");
                                }

                                void CTPCC_OLEDB::Delivery()
                                {
                                HRESULT
                                hr;
                                int
                                iTryCount = 0;
                                IRowset*
                                pRowset;
                                LONG
                                cRows = 1; // number of rows
                                returned in the rowset
                                ULONG
                                cRowsObtained;
                                HROW
                                rghRow;
                                //returned row handles
                                HROW*
                                prghRow = &rghRow;

                                while (TRUE)
                                {
                                try
                                {
                                //
                                Execute the prepared command
                                hr =
                                m_pIDeliveryCommand->Execute(NULL,
                                IID_IRowset, &m_DeliveryExecuteParams, NULL,
                                (IUnknown
                                **) &pRowset);
                                if
                                (FAILED(hr))
                                {
                                ThrowError(m_pIDeliveryCommand,
                                COLEDBERR::eExecute, "Delivery()");
                                }

                                // Fetch
                                the result row handle(s)
                                hr =
                                pRowset->GetNextRows(DB_NULL_HCHAPTER,
                                0, cRows, &cRowsObtained, &prghRow);
                                if
                                (FAILED(hr))
                                {
                                ThrowError(m_pIDeliveryCommand,
                                COLEDBERR::eGetNextRows, "Delivery()");
                                }

                                // Fetch
                                the actual row data by handle
                                hr =
                                pRowset->GetData(rghRow,
                                m_hDeliveryOutputAccessor, &m_txn.Delivery);
                                if
                                (FAILED(hr))

```

```

    {
        ThrowError(m_pIDeliveryCommand,
        COLEDBERR::eGetData, "Delivery()");
    }
    //
    Release row(s)
    hr =
    pRowset->ReleaseRows(cRowsObtained,
    prghRow, NULL, NULL, NULL);
    //
    Release rowset
    hr =
    pRowset->Release();

    m_txn.Delivery.exec_status_code =
    eOK;

    break;
    }
    catch (COLEDBERR
    *e)
    {
        if (!(!e-
        >m_bDeadLock) || (++iTryCount >
        iMaxRetries))
            throw;

        // hit
        deadlock; backoff for increasingly longer period
        delete e;
        Sleep(10

    * iTryCount);
    }

    //
    // if (iTryCount)
    // throw new
    CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_R
    ETRIED_TRANS, iTryCount);
    }

```

tm_com_dll\src\tpcc_com.cpp

```

/* FILE:
   TPCC_COM.CPP
*
* Microsoft TPC-C Kit Ver. 4.20.000
*
* Copyright Microsoft, 1999
* All Rights Reserved
*
* not yet audited
*
* PURPOSE: Source file for TPC-C
COM+ class implementation.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - first
version
*/
// needed for CoInitializeEx
#define _WIN32_WINNT 0x0400

```

```

#include <windows.h>
// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
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
    =

    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
m_pPayment =
same component
m_pNewOrder;

m_pStockLevel =
m_pNewOrder;

m_pOrderStatus =
m_pNewOrder;
}
else
{
    // use different
    components for each txn

    hr =
    CoCreateInstance(CLSID_NewOrder, NULL,
    CLSCTX_SERVER, IID_ITPCC, (void
    **)&m_pNewOrder);
    if (FAILED(hr))
        throw

    new CCOMERR(hr);

    hr =
    CoCreateInstance(CLSID_Payment, NULL,
    CLSCTX_SERVER, IID_ITPCC, (void
    **)&m_pPayment);
    if (FAILED(hr))
        throw

    new CCOMERR(hr);

    hr =
    CoCreateInstance(CLSID_StockLevel, NULL,
    CLSCTX_SERVER, IID_ITPCC, (void
    **)&m_pStockLevel);
    if (FAILED(hr))
        throw

    new CCOMERR(hr);

    hr =
    CoCreateInstance(CLSID_OrderStatus, NULL,
    CLSCTX_SERVER, IID_ITPCC, (void
    **)&m_pOrderStatus);
    if (FAILED(hr))
        throw

    new CCOMERR(hr);

    // call setcomplete to release each
    component back into pool
    hr = m_pNewOrder-
    >CallSetComplete();
    if (FAILED(hr))
        throw new
    CCOMERR(hr);

    if (!m_bSinglePool)
    {
        hr = m_pPayment-
        >CallSetComplete();
        if (FAILED(hr))
            throw

        new CCOMERR(hr);

        hr = m_pStockLevel-
        >CallSetComplete();
        if (FAILED(hr))
            throw

        new CCOMERR(hr);

        hr = m_pOrderStatus-
        >CallSetComplete();
        if (FAILED(hr))
            throw

        new CCOMERR(hr);
    }
}

```



```

CTPCC_COM::~~CTPCC_COM()
{
    if ( m_pTxn )
        SafeArrayDestroy(m_vTxn.parray);

    ReleaseInterface(m_pNewOrder);
    if (!m_bSinglePool)
    {
        ReleaseInterface(m_pPayment);
        ReleaseInterface(m_pStockLevel);
        ReleaseInterface(m_pOrderStatus);
    }
    CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pNewOrder-
>NewOrder(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray)
;

    if ( m_pTxn->ErrorType !=
ERR_SUCCESS )
        throw new CCOMERR(
m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::Payment()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pPayment-
>Payment(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray)
;

    if ( m_pTxn->ErrorType !=
ERR_SUCCESS )
        throw new CCOMERR(
m_pTxn->ErrorType, m_pTxn->error );
}

void CTPCC_COM::StockLevel()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pStockLevel-
>StockLevel(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
}

```

```

        SafeArrayDestroy(vTxn_out.parray)
;
        if ( m_pTxn->ErrorType !=
ERR_SUCCESS )
            throw new CCOMERR(
m_pTxn->ErrorType, m_pTxn->error );
    }
}

void CTPCC_COM::OrderStatus()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pOrderStatus-
>OrderStatus(m_vTxn, &vTxn_out);
    if (FAILED(hr))
        throw new CCOMERR(
hr );
    memcpy(m_pTxn, (void
*)vTxn_out.parray->pvData,vTxn_out.parray-
>rgsabound[0].cElements);
    SafeArrayDestroy(vTxn_out.parray)
;

    if ( m_pTxn->ErrorType !=
ERR_SUCCESS )
        throw new CCOMERR(
m_pTxn->ErrorType, m_pTxn->error );
}

```

\tpcc_com_all\src\methods.h

```

/*      FILE:
        METHODS.H

        *      Microsoft TPC-C Kit Ver. 4.20.000

        *      Copyright Microsoft, 1999
        All Rights Reserved

        *      not yet audited

        *      PURPOSE:  Header file for COM
        components.

        *      Change history:
        *      4.20.000 - first
        version
        */

enum COMPONENT_ERROR
{
    ERR_MISSING_REGISTRY_ENTRIES
= 1,
    ERR_LOADDLL_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
public:
    CCOMPONENT_ERR(COMPONENT_
ERROR Err)
    {
        m_Error
= Err;
}
}

```

```

        m_szTextDetail = NULL;
        m_SystemErr = 0;

        m_szErrorText = NULL;
    };

    CCOMPONENT_ERR(COMPONENT_
ERROR Err, char *szTextDetail, DWORD
dwSystemErr)
    {
        m_Error
= Err;

        m_szTextDetail = new
char[strlen(szTextDetail)+1];
        strcpy(
m_szTextDetail, szTextDetail );

        m_SystemErr = dwSystemErr;

        m_szErrorText = NULL;
    };

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

    COMPONENT_ERROR
m_Error;
    char
*m_szTextDetail;
    char
*m_szErrorText;
    DWORD
m_SystemErr;

    int ErrorType()
{return ERR_TYPE_COMPONENT;};
    char *ErrorTypeStr() {
return "COMPONENT"; }
    int ErrorNum() {return
m_Error;};
    char *ErrorText();
};

static void WriteMessageToEventLog(LPTSTR
lpszMsg);

////////////////////////////////////
////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
public ITPCC,
public IObjectControl,
public IObjectConstruct,
public
CComObjectRootEx<CComSingleThreadModel>
{
public:
    BEGIN_COM_MAP(CTPCC_Common)
        COM_INTERFACE_ENTRY(ITPCC)
}

```

```

COM_INTERFACE_ENTRY(IObjectC
ontrol)
COM_INTERFACE_ENTRY(IObjectC
onstruct)
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
    STDMETHODCALLTYPE (BOOL)
    CanBePooled() { return m_bCanBePooled; }
    STDMETHODCALLTYPE (void)
    S_OK; } // we don't support COM Services
    transactions (no enlistment)
    STDMETHODCALLTYPE (void)
    Deactivate() { /* nothing to do */ }

// IObjectConstruct
    STDMETHODCALLTYPE
    Construct(IDispatch * pUnk);

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

    struct COM_DATA
    {
        int retval;
        int error;
        union
        {
            NEW_ORDER_DATA
            NewOrder;

            PAYMENT_DATA
            Payment;

            DELIVERY_DATA
            Delivery;

            STOCK_LEVEL_DATA
            StockLevel;

            ORDER_STATUS_DATA
            OrderStatus;
        } u;
    };
};

```

```

////////////////////////////////////
//ITPCC////////////////////////////////
class CTPCC :
    public CTPCC_Common,
    public CComCoClass<CTPCC,
    &CLSID_TPCC>
    {
    public:
    DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

    BEGIN_COM_MAP(CTPCC)
        //COM_INTERFACE_ENTRY2(IUnkno
        own,
        CComObjectRootEx<CComSingleThreadModel>)
        COM_INTERFACE_ENTRY2(IUnkno
        wn, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(C
        TPCC_Common)
    END_COM_MAP()
    };

////////////////////////////////////
////////////////////////////////////
// CNewOrder
class CNewOrder :
    public CTPCC_Common,
    public CComCoClass<CNewOrder,
    &CLSID_NewOrder>
    {
    public:
    DECLARE_REGISTRY_RESOURCEID(IDR_NEWOR
    DER)

    BEGIN_COM_MAP(CNewOrder)
        //
        COM_INTERFACE_ENTRY2(IUnkno
        wn, CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnkno
        wn, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(C
        TPCC_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_ORDER
    STATUS)

    BEGIN_COM_MAP(COrderStatus)

```

```

//
        COM_INTERFACE_ENTRY2(IUnkno
        wn, CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnkno
        wn, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(C
        TPCC_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_PAYME
    NT)

    BEGIN_COM_MAP(CPayment)
        //
        COM_INTERFACE_ENTRY2(IUnkno
        wn, CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnkno
        wn, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(C
        TPCC_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_STOCKL
    EVEL)

```

```

BEGIN_COM_MAP(CStockLevel)
//
//      COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
//      COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//      COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in,
        VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in,
        VARIANT* txn_out) {return E_NOTIMPL;}
    //      HRESULT __stdcall StockLevel(
    //          VARIANT txn_in, VARIANT*
    //          txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT*
        txn_out) {return E_NOTIMPL;}
};

```

```

\tpcc com all\src\resource.
h

```

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include
// file.
// Used by tpcc_com_all.rc
//
#define IDS_PROJNAME      100
#define IDR_TPCC          101
#define IDR_NEWORDER     102
#define IDR_ORDERSTATUS  103
#define IDR_PAYMENT      104
#define IDR_STOCKLEVEL   105

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE  202
#define _APS_NEXT_COMMAND_VALUE  32768
#define _APS_NEXT_CONTROL_VALUE   201
#define _APS_NEXT_SYMED_VALUE    106
#endif
#endif

```

```

\tpcc com all\src\tpcc com
all.cpp

```

```

/*      FILE:
        TPCC_COM_ALL.CPP

        *      Microsoft TPC-C Kit Ver. 4.20.000

        *      Copyright Microsoft, 1999

```

```

*      All Rights Reserved
*
*      Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
*
*      PURPOSE:  Implementation for
TPC-C Tuxedo class.
*      Contact:  Charles Levine
(clevine@microsoft.com)
*
*      Change history:
*      4.20.000 - updated
rev number to match kit
*/

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <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(hInstanc
e);

            DWORD
dwSize = MAX_COMPUTERNAME_LENGTH+1;
            GetComputerName(szMyComputerN
ame, &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(&hConnectC
                            riticalSection);

                            }

                            }
                            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("U
        nhandled 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
                (LPTSTR *)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* pObjContext =
    NULL;

    // get our object context
    HRESULT hr =
    CoGetObjectContext( IID_IObjectContext, (void
    ***)&pObjContext );

```

```

    ReleaseInterface(pObjContext);
    return hr;
}

//
// called by the ctor activator
//
//
// STDMETHODCALLTYPE
// 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];

```

```

        _snprintf(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("U
        nhandled 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_U11,
        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("U
nhandled 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("U
nhandled 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("U
nhandled 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("U
nhandled exception.));
        pData->retval =
ERR_TYPE_LOGIC;
        pData->error = 0;
        m_bCanBePooled =
FALSE;
        return E_FAIL;
    }
}

```

```

\tpcc com all\src\tpcc com
all.def

```

```

/*      FILE:
        TPCC_COM_ALL.CPP
*
*      Microsoft TPC-C Kit Ver. 4.20.000
*
*      Copyright Microsoft, 1999
*      All Rights Reserved
*
*      Version 4.10.000 audited by
Richard Gimarc, Performance Metrics, 3/17/99
*
*      PURPOSE:  Implementation for
TPC-C Tuxedo class.
*      Contact:  Charles Levine
(clevine@microsoft.com)
*
*      Change history:
*      4.20.000 - updated
rev number to match kit
*/

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>

```

```

// 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_COMPUT
ERNAME_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(hInstanc
e);

            DWORD
dwSize = MAX_COMPUTERNAME_LENGTH+1;
            GetComputerName(szMyComputerN
ame, &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(&hConnectC
riticalSection);
        }

        }
        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("U
nhandled 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
            (LPTSTR *)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].i.Error)
        {
            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

//
// STDMETHODCALLTYPE
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("U
nhandled 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("U
nhandled 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("U
nhandled 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-

        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();

```

```

        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("U
nhandled 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-

        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("U
nhandled 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*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__
440
#endif

```

```

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

```

```

#ifdef __tpcc_com_all_h__
#define __tpcc_com_all_h__

```

```

#if defined(_MSC_VER) && (_MSC_VER >=
1020)
#pragma once
#endif

```

```

/* Forward Declarations */

```

```

#ifdef __TPCC_FWD_DEFINED__

```

```

#define __TPCC_FWD_DEFINED__
#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

```

```

#ifdef __TPCC_FWD_DEFINED__ */

```

```

#ifdef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

```

```

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

```

```

#ifdef __NewOrder_FWD_DEFINED__
*/

```

```

#ifdef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

```

```

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

```

```

#ifdef __OrderStatus_FWD_DEFINED__
*/

```

```

#ifdef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

```

```

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

```

```

#ifdef __Payment_FWD_DEFINED__ */

```

```

#ifdef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

```

```

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

```

```

#ifdef __StockLevel_FWD_DEFINED__
*/

```

```

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

```

```

#ifdef __cplusplus
extern "C"{
#endif

```

```

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

```

```

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

```

```

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

```

```

#ifdef __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

```

```

#ifdef __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;
    };
};

```

```
\tpcc com all\src\tpcc com
all.rc
```

```

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

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

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

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

```

```

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

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

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

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

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
    "\0"
END

#endif // APSTUDIO_INVOKED

#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
END

```

```

BLOCK "VarFileInfo"
BEGIN
    VALUE "Translation", 0x409, 1200
END
END

#endif // !_MAC

////////////////////////////////////
////////////////////////////////////
//
// REGISTRY
//
IDR_TPCC            REGISTRY DISCARDABLE
"tpcc_com_all.rgs"
IDR_NEWORDER       REGISTRY
DISCARDABLE "tpcc_com_no.rgs"
IDR_ORDERSTATUS   REGISTRY
DISCARDABLE "tpcc_com_os.rgs"
IDR_PAYMENT        REGISTRY DISCARDABLE
"tpcc_com_pay.rgs"
IDR_STOCKLEVEL    REGISTRY
DISCARDABLE "tpcc_com_sl.rgs"

////////////////////////////////////
////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME        "tpcc_com_all"
END

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

#ifndef APSTUDIO_INVOKED
////////////////////////////////////
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

////////////////////////////////////
////////////////////////////////////
#endif // not APSTUDIO_INVOKED

\tpcc com all\src\tpcc com
all.rgs

HKCR
{
    TPCC.AITXns.1 = s 'All Txns Class'
    {
        CLSID = s
        '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
    }
    TPCC.AITXns = s 'TPCC Class'
    {
        CurVer = s
        'TPCC.AITXns.1'
    }
}

```

```

NoRemove CLSID
ForceRemove
{122A3128-2520-11D3-BA71-00C04FBFE08B} =
s 'TPCC Class'
{
    ProgID
    = s 'TPCC.AITXns.1'

    VersionIndependentProgID = s
    'TPCC.AITXns'

    InprocServer32 = s '%MODULE%'
    {
        val ThreadingModel = s 'Both'
    }
}

#pragma warning( disable: 4049 ) /* more than
64k source lines */

/* this ALWAYS GENERATED file contains the
IDs and CLSIDs */

/* link this file in with the server and any clients
*/

/* File created by MIDL compiler version
6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check
enum stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifndef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#endif


```

```

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b
3,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,b
3,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,0x122A3117,0x2520,0x11D3,0xB
A,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0
x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0
xBA,0x47,0x00,0xC0,0x4F,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,0x11D2,0x
BA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0
xBA,0x4E,0x00,0xC0,0x4F,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_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,b
3,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,b
3,b4,b5,b6,b7,b8) name =
{l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#ifdef !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xB
A,0x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0
x71,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0
xBA,0x47,0x00,0xC0,0x4F,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,0x11D2,0x
BA,0x4E,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0
xBA,0x4E,0x00,0xC0,0x4F,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(notable)
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

/* Forward Declarations */

#ifdef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif /* __ITPCC_FWD_DEFINED__ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

```

```

extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

```

```

#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("FEE6AA2-84B1-11d2-
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(

```

```

/* [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 __stdcall ITPCC_NewOrder_Proxy(
ITPCC * This,
/* [in] */ VARIANT txn_in,
/* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Delivery_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,

```

```

DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_StockLevel_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_OrderStatus_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall
ITPCC_CallSetComplete_Proxy(
ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *_pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

#endif /*
__ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long __RPC_USER
VARIANT_UserSize( unsigned long *, unsigned
long , VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserMarshal( unsigned long *,
unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal( unsigned long *,
unsigned char *, VARIANT * );
void __RPC_USER
VARIANT_UserFree( unsigned long *,
VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif


```

```

\tpccc com all\src\tpccc com
sl.rgs

```

```

HKCR
{
    TPCC.StockLevel.1 = s 'StockLevel
Class'
    {
        CLSID = s
        '{2668369E-A50D-11D2-BA4E-00C04FBFE08B}'
    }
    TPCC.StockLevel = s 'StockLevel
Class'
    {
        CurVer = s
        'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove
        {2668369E-A50D-11D2-BA4E-00C04FBFE08B} =
s 'StockLevel Class'
    }
    ProgID
    = s 'TPCC.StockLevel.1'
    VersionIndependentProgID = s
    'TPCC.StockLevel'
    InprocServer32 = s '%MODULE%'
    {
        val ThreadingModel = s 'Both'
    }
}

```

```

\tpccc_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( tpccc_com_ps )

PROXYFILE_LIST_START
/* Start of list */
REFERENCE_PROXY_FILE( tpccc_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*/
```

```
#ifndef __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__
```

```
#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;
```

```
#if defined(__cplusplus) &&
!defined(CINTERFACE)
```

```
MIDL_INTERFACE("FEE6AA2-84B1-11d2-
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 )
```

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

```
HRESULT ( STDMETHODCALLTYPE )
```

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

```
HRESULT ( STDMETHODCALLTYPE )
```

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

```
HRESULT ( STDMETHODCALLTYPE )
```

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

```
HRESULT ( STDMETHODCALLTYPE )
```

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

```
HRESULT ( STDMETHODCALLTYPE )
```

```
ITPCC * This);
```

```
END_INTERFACE
```

```
} ITPCCVtbl;
```

```
interface ITPCC
```

```
{
CONST_VTBL struct ITPCCVtbl *lpVtbl;
```

```

#ifndef 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 */

/* C style interface */

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

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

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

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall
ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall
ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

#endif /*
__ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long __RPC_USER
VARIANT_UserSize( unsigned long *, unsigned
long
, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserMarshal( unsigned long *,
unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal( unsigned long *,
unsigned char *, VARIANT * );
void __RPC_USER
VARIANT_UserFree( unsigned long *,
VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}

```

```

#endif
#endif

tpcc com ps\src\tpcc com ps
.idl

FILE:
ITPCC.IDL

*
Microsoft TPC-C Kit Ver. 4.20.000

*
Copyright Microsoft, 1999
All Rights Reserved

*
not yet audited

*
PURPOSE: Defines the interface
used by TPCC. This interface can be implemented
by C++ components.
*
* Change history:
*
* version 4.20.000 - first
*/

// 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
(
    VARIANT txn_in,
    VARIANT *txn_out
);

    HRESULT __stdcall Payment
(
    VARIANT txn_in,
    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(5066b1b), 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,b
3,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,b
3,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,0x
47,0x00,0xC0,0x4F,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 14:48:43 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,b
3,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,0x
47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif
```

```
#endif /* defined(_M_IA64) ||
defined(_M_AMD64) */
```

```
tpcc com ps\src\tpcc com ps
_p.c
```

```
#pragma warning( disable: 4049 ) /* more than
64k source lines */
```

```
/* this ALWAYS GENERATED file contains the
proxy stub code */
```

```
/* File created by MIDL compiler version
6.00.0347 */
```

```
/* at Fri Apr 15 14:48:43 2005
```

```
*/
```

```
/* Compiler settings for .\src\tpcc com ps.idl:
```

```
Oicf, W1, Zp8, env=Win32 (32b run)
```

```
protocol : dce , ms_ext, c_ext
```

```
error checks: allocation ref bounds_check
```

```
enum stub_data
```

```
VC __declspec() decoration level:
```

```
__declspec(uuid()), __declspec(selectany),
```

```
__declspec(novtable)
```

```
DECLSPEC_UUID(), MIDL_INTERFACE()
```

```
*/
```

```
//@@MIDL_FILE_HEADING( )
```

```
#if !defined(_M_IA64) && !defined(_M_AMD64)
```

```
#define USE_STUBLESS_PROXY
```

```
/* verify that the <rpcproxy.h> version is high
enough to compile this file*/
```

```
#ifndef __REDQ_RPCPROXY_H_VERSION__
```

```
#define __REQUIRED_RPCPROXY_H_VERSION__
```

```
440
```

```
#endif
```

```
#include "rpcproxy.h"
```

```
#ifndef __RPCPROXY_H_VERSION__
```

```
#error this stub requires an updated version of
```

```
<rpcproxy.h>
```

```
#endif // __RPCPROXY_H_VERSION__
```

```
#include "tpcc_com_ps.h"
#define TYPE_FORMAT_STRING_SIZE 1023
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1
```

```
typedef struct _MIDL_TYPE_FORMAT_STRING
{
short Pad;
unsigned char Format[
TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;
```

```
typedef struct _MIDL_PROC_FORMAT_STRING
{
short Pad;
unsigned char Format[
PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;
```

```
static RPC_SYNTAX_IDENTIFIER
_RpcTransferSyntax =
{{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,
0x00,0x2B,0x10,0x48,0x60}},{2,0}};
```

```
extern const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString;
```

```
extern const MIDL_STUB_DESC
Object_StubDesc;
```

```
extern const MIDL_SERVER_INFO
ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;
```

```
extern const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ];
```

```
#if !defined(_RPC_WIN32_)
#error Invalid build platform for this stub.
#endif
```

```
#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to
run this stub because it uses these features:
```

```
#error -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 */
```

```
0x33,
/* FC_AUTO_HANDLE
0x6C,
*/
/* Old Flags: object,
```

```
Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
/* 8 */ NdrFcShort( 0x1c ), /* x86
Stack size/offset = 28 */
```

```
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2
```

```
Flags: srv must size, clt must size, has return, */
0x3,
/* 3 */
```

```
/* Parameter txn_in */
```

```
/* 16 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 18 */ NdrFcShort( 0x4 ), /* x86
```

```
Stack size/offset = 4 */
/* 20 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */
```

```
/* Parameter txn_out */
```

```
/* 22 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
```

```
/* 24 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 26 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */
```

```
/* Return value */
```

```
/* 28 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 30 */ NdrFcShort( 0x18 ), /* x86
```

```
Stack size/offset = 24 */
/* 32 */ 0x8, /*
FC_LONG */
```

```
0x0,
```

```
/* 0 */
```

```
/* Procedure Payment */
```

```
/* 34 */ 0x33, /*
FC_AUTO_HANDLE */
```

```
0x6C,
/* Old Flags: object,
```

```
Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x1c ), /* x86
```

```
Stack size/offset = 28 */
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2
```

```
Flags: srv must size, clt must size, has return, */
0x3,
```

```
/* 3 */
```

```
/* Parameter txn_in */
```

```
/* 50 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 52 */ NdrFcShort( 0x4 ), /* x86
```

```
Stack size/offset = 4 */
/* 54 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */
```

```
/* Parameter txn_out */
```

```

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 58 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 60 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 64 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 66 */ 0x8, /*
FC_LONG */

/* 0 */

/* Procedure Delivery */

/* 68 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
0x6c,
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, */
/* 3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags:
must size, must free, in, by val, */
/* 86 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 88 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 92 */ NdrFcShort( 0x14 ), /* x86
Stack size/offset = 20 */
/* 94 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 98 */ NdrFcShort( 0x18 ), /* x86
Stack size/offset = 24 */
/* 100 */ 0x8, /*
FC_LONG */

/* 0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
0x6c,
Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */

```

```

/* 100 */ NdrFcShort( 0x8c ), /* 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, */
/* 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 */

/* 0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
0x6c,
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, */
/* 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 */

/* 0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
0x6c,
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, */
/* 1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags:
out, return, base type, */
/* 188 */ NdrFcShort( 0x4 ), /* x86
Stack size/offset = 4 */
/* 190 */ 0x8, /*
FC_LONG */

/* 0 */

/* 0 */

}
};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
/* 2 */
0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3ca ), /*
Offset= 970 (974) */
/* 6 */
0x2b,

/*
FC_NON_ENCAPSULATED_UNION */
0x9,
/* FC_ULONG */
/* 8 */ 0x7, /* Corr
desc: FC_USHORT */
0x0,

/* */
/* 10 */ NdrFcShort( 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_FLOAT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x800a ), /*
Simple arm type: FC_LONG */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 54 */ NdrFcLong( 0xb ), /* 11 */
/* 58 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_FLOAT */
/* 60 */ NdrFcLong( 0xa ), /* 10 */
/* 64 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 66 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /*
Offset= 232 (302) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 76 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 78 */ NdrFcLong( 0x8 ), /* 8 */
/* 82 */ NdrFcShort( 0xe2 ), /*
Offset= 226 (308) */
/* 84 */ NdrFcLong( 0xd ), /* 13 */
/* 88 */ NdrFcShort( 0xf4 ), /*
Offset= 244 (332) */
/* 90 */ NdrFcLong( 0x9 ), /* 9 */
/* 94 */ NdrFcShort( 0x100 ), /*
Offset= 256 (350) */
/* 96 */ NdrFcLong( 0x2000 ), /* 8192
*/
/* 100 */ NdrFcShort( 0x10c ), /*
Offset= 268 (368) */
/* 102 */ NdrFcLong( 0x24 ), /* 36 */
/* 106 */ NdrFcShort( 0x31a ), /*
Offset= 794 (900) */
/* 108 */ NdrFcLong( 0x4024 ), /* 16420
*/
/* 112 */ NdrFcShort( 0x314 ), /*
Offset= 788 (900) */
/* 114 */ NdrFcLong( 0x4011 ), /* 16401
*/
/* 118 */ NdrFcShort( 0x312 ), /*
Offset= 786 (904) */
/* 120 */ NdrFcLong( 0x4002 ), /* 16386
*/
/* 124 */ NdrFcShort( 0x310 ), /*
Offset= 784 (908) */
/* 126 */ NdrFcLong( 0x4003 ), /* 16387
*/
/* 130 */ NdrFcShort( 0x30e ), /*
Offset= 782 (912) */
/* 132 */ NdrFcLong( 0x4014 ), /* 16404
*/
/* 136 */ NdrFcShort( 0x30c ), /*
Offset= 780 (916) */
/* 138 */ NdrFcLong( 0x4004 ), /* 16388
*/
/* 142 */ NdrFcShort( 0x30a ), /*
Offset= 778 (920) */
/* 144 */ NdrFcLong( 0x4005 ), /* 16389
*/
/* 148 */ NdrFcShort( 0x308 ), /*
Offset= 776 (924) */
/* 150 */ NdrFcLong( 0x400b ), /* 16395
*/
/* 154 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (908) */

```

```

/* 156 */ NdrFcLong( 0x400a ), /* 16394
*/
/* 160 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (912) */
/* 162 */ NdrFcLong( 0x4006 ), /* 16390
*/
/* 166 */ NdrFcShort( 0x2fa ), /*
Offset= 762 (928) */
/* 168 */ NdrFcLong( 0x4007 ), /* 16391
*/
/* 172 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (924) */
/* 174 */ NdrFcLong( 0x4008 ), /* 16392
*/
/* 178 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (932) */
/* 180 */ NdrFcLong( 0x400d ), /* 16397
*/
/* 184 */ NdrFcShort( 0x2f0 ), /*
Offset= 752 (936) */
/* 186 */ NdrFcLong( 0x4009 ), /* 16393
*/
/* 190 */ NdrFcShort( 0x2ee ), /*
Offset= 750 (940) */
/* 192 */ NdrFcLong( 0x6000 ), /* 24576
*/
/* 196 */ NdrFcShort( 0x2ec ), /*
Offset= 748 (944) */
/* 198 */ NdrFcLong( 0x400c ), /* 16396
*/
/* 202 */ NdrFcShort( 0x2ea ), /*
Offset= 746 (948) */
/* 204 */ NdrFcLong( 0x10 ), /* 16
*/
/* 208 */ NdrFcShort( 0x8002 ), /*
Simple arm type: FC_CHAR */
/* 210 */ NdrFcLong( 0x12 ), /* 18
*/
/* 214 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 216 */ NdrFcLong( 0x13 ), /* 19
*/
/* 220 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0x15 ), /* 21
*/
/* 226 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 228 */ NdrFcLong( 0x16 ), /* 22
*/
/* 232 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 234 */ NdrFcLong( 0xe ), /* 23
*/
/* 238 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 240 */ NdrFcLong( 0xe ), /* 24
*/
/* 244 */ NdrFcShort( 0x2c8 ), /*
Offset= 712 (956) */
/* 246 */ NdrFcLong( 0x400e ), /* 16398
*/
/* 250 */ NdrFcShort( 0x2cc ), /*
Offset= 716 (966) */
/* 252 */ NdrFcLong( 0x4010 ), /* 16400
*/
/* 256 */ NdrFcShort( 0x2ca ), /*
Offset= 714 (970) */
/* 258 */ NdrFcLong( 0x4012 ), /* 16402
*/
/* 262 */ NdrFcShort( 0x286 ), /*
Offset= 646 (908) */
/* 264 */ NdrFcLong( 0x4013 ), /* 16403
*/
/* 268 */ NdrFcShort( 0x284 ), /*
Offset= 644 (912) */
/* 270 */ NdrFcLong( 0x4015 ), /* 16405
*/
/* 274 */ NdrFcShort( 0x282 ), /*
Offset= 642 (916) */
/* 276 */ NdrFcLong( 0x4016 ), /* 16406
*/

```

```

/* 280 */ NdrFcShort( 0x278 ), /*
Offset= 632 (912) */
/* 282 */ NdrFcLong( 0x4017 ), /*
*/
/* 286 */ NdrFcShort( 0x272 ), /*
Offset= 626 (912) */
/* 288 */ NdrFcLong( 0x0 ), /* 0
*/
/* 292 */ NdrFcShort( 0x0 ), /*
Offset= 0 (292) */
/* 294 */ NdrFcLong( 0x1 ), /* 1
*/
/* 298 */ NdrFcShort( 0x0 ), /*
Offset= 0 (298) */
/* 300 */ NdrFcShort( 0xffffffff ), /*
Offset= -1 (299) */
/* 302 */
/* 304 */ NdrFcShort( 0x8 ), /* 8
*/
/* 306 */ 0xb, /*
FC_HYPER */
/* 308 */
/* 310 */ NdrFcShort( 0xc ), /*
Offset= 12 (322) */
/* 312 */
/* 314 */ NdrFcShort( 0x2 ), /* 2
*/
/* 316 */ 0x9, /*
desc: FC_ULONG */
/* 318 */ NdrFcShort( 0xffc ), /* -4
*/
/* 320 */ 0x6, /*
FC_SHORT */
/* 322 */
/* 324 */ NdrFcShort( 0x8 ), /* 8
*/
/* 326 */ NdrFcShort( 0xfffff2 ), /*
Offset= -14 (312) */
/* 328 */ 0x8, /*
FC_LONG */
/* 330 */ 0x5c, /*
FC_PAD */
/* 332 */
/* 334 */ NdrFcLong( 0x0 ), /* 0
*/
/* 338 */ NdrFcShort( 0x0 ), /* 0
*/
/* 340 */ NdrFcShort( 0x0 ), /* 0
*/
/* 342 */ 0xc0, /* 192
*/
/* 344 */ 0x0, /* 0
*/

```

```

/* 346 */ 0x0, /* 0 */
/* 348 */ 0x0, /* 0 */
/* 350 */ /* 70 */
/* 352 */ NdrFcLong( 0x20400 ), /*
132096 */
/* 356 */ NdrFcShort( 0x0 ), /* 0 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* 360 */ 0xc0, /* 192 */
/* 362 */ 0x0, /* 0 */
/* 364 */ 0x0, /* 0 */
/* 366 */ 0x0, /* 0 */
/* 368 */ /* 70 */
0x10, /* FC_UP [pointer_deref] */
/* 370 */ NdrFcShort( 0x2 ), /*
Offset= 2 (372) */
/* 372 */
0x0, /* FC_UP */
/* 374 */ NdrFcShort( 0x1fc ), /*
Offset= 508 (882) */
/* 376 */
FC_ENCAPSULATED_UNION /*
/* 378 */ NdrFcShort( 0x18 ), /* 24 */
/* 380 */ NdrFcShort( 0xa ), /* 10 */
/* 382 */ NdrFcLong( 0x8 ), /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /*
Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ), /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /*
Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ), /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /*
Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ), /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /*
Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ), /* 36 */
/* 410 */ NdrFcShort( 0x114 ), /*
Offset= 276 (686) */
/* 412 */ NdrFcLong( 0x800d ), /* 32781 */
/* 416 */ NdrFcShort( 0x130 ), /*
Offset= 304 (720) */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /*
Offset= 328 (750) */
/* 424 */ NdrFcLong( 0x2 ), /* 2 */
/* 428 */ NdrFcShort( 0x160 ), /*
Offset= 352 (780) */
/* 430 */ NdrFcLong( 0x3 ), /* 3 */
/* 434 */ NdrFcShort( 0x178 ), /*
Offset= 376 (810) */

/* 440 */ NdrFcShort( 0x140 ), /* 20 */
Offset= 400 (840) */
/* 442 */ NdrFcShort( 0xfffff ), /*
Offset= -1 (441) */
/* 444 */
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
/* 454 */
FC_VARIABLE_REPEAT /*
/* 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( 0xfffff6e ), /*
Offset= -146 (322) */
/* 470 */
/* 472 */ 0x5c, /* FC_LONG */
/* 474 */
/* 476 */ NdrFcShort( 0x8 ), /* 8 */
/* 478 */
/* 480 */
/* 482 */ NdrFcShort( 0x4 ), /* 4 */
/* 484 */ NdrFcShort( 0x4 ), /* 4 */
/* 486 */ 0x11, 0x0, /* FC_RP */
/* 488 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (444) */
/* 490 */
/* 492 */ 0x8, /* FC_LONG */
FC_LONG */

/* 494 */ /* FC_END */
/* 496 */ NdrFcShort( 0x0 ), /* 0 */
/* 498 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 500 */ NdrFcShort( 0x0 ), /* 0 */
/* 502 */ NdrFcLong( 0xfffff ), /* -1 */
/* 506 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 508 */ NdrFcShort( 0xfffff50 ), /*
Offset= -176 (332) */
/* 510 */ 0x5c, /*
FC_PAD */
/* 512 */ /* FC_END */
FC_BOGUS_STRUCT /*
/* 514 */ NdrFcShort( 0x8 ), /* 8 */
/* 516 */ NdrFcShort( 0x0 ), /* 0 */
/* 518 */ NdrFcShort( 0x6 ), /*
Offset= 6 (524) */
/* 520 */ 0x8, /*
FC_LONG */
/* 522 */ 0x5c, /* FC_POINTER */
/* 524 */ /* FC_END */
/* 526 */ NdrFcShort( 0xfffff0 ), /*
Offset= -32 (494) */
/* 528 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xfffff ), /* -1 */
/* 540 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 542 */ NdrFcShort( 0xfffff40 ), /*
Offset= -192 (350) */
/* 544 */ 0x5c, /*
FC_PAD */
/* 546 */ /* FC_END */
FC_BOGUS_STRUCT */

```

```

/* 3 */ 0x3,
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /*
Offset= 6 (558) */
/* 554 */ 0x8, /*
FC_LONG */

/* 556 */ 0x5c, /* FC_POINTER */
FC_PAD */

/* 558 */ /* FC_END */

/* 560 */ /* FC_RP */
/* 560 */ NdrFcShort( 0xfffff0 ), /*
Offset= -32 (528) */
/* 562 */

/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */

/* 572 */ /* FC_PP */
/* FC_PAD */

/* 574 */ /* FC_END */

FC_VARIABLE_REPEAT */

/* 576 */ /* FC_FIXED_OFFSET */
/* 574 */ NdrFcShort( 0x4 ), /* 4 */
/* 576 */ NdrFcShort( 0x0 ), /* 0 */
/* 578 */ NdrFcShort( 0x1 ), /* 1 */
/* 580 */ NdrFcShort( 0x0 ), /* 0 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ 0x12, 0x0, /* FC_UP */
/* 586 */ NdrFcShort( 0x184 ), /*
Offset= 388 (974) */
/* 588 */

/* 590 */ 0x5c, /* FC_END */
FC_PAD */

/* 592 */ /* FC_LONG */
/* FC_END */

FC_BOGUS_STRUCT */

/* 594 */ NdrFcShort( 0x8 ), /* 8 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x6 ), /*
Offset= 6 (604) */
/* 600 */ 0x8, /*
FC_LONG */

/* 602 */ /* FC_POINTER */

```

```

/* 602 */ 0x5c, /*
FC_PAD */

/* 604 */ /* FC_END */

/* 606 */ /* FC_RP */
/* 606 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (562) */
/* 608 */

/* 610 */ NdrFcLong( 0x2f ), /* 47 */
/* 614 */ NdrFcShort( 0x0 ), /* 0 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ 0xc0, /* 192 */

/* 620 */ 0x0, /* 0 */

/* 622 */ 0x0, /* 0 */

/* 624 */ 0x0, /* 0 */
/* 626 */ /* 70 */

/* 628 */ NdrFcShort( 0x1 ), /* 1 */
/* 630 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 632 */ NdrFcShort( 0x4 ), /* 4 */
/* 634 */ 0x1, /*
FC_BYTE */

/* 636 */ /* FC_END */

FC_BOGUS_STRUCT */

/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0xa ), /*
Offset= 10 (652) */
/* 644 */ 0x8, /*
FC_LONG */

/* 646 */ 0x4c, /* FC_LONG */
FC_EMBEDDED_COMPLEX */

/* 648 */ NdrFcShort( 0xfffffd8 ), /*
Offset= -40 (608) */
/* 650 */ 0x36, /*
FC_POINTER */

/* 652 */ /* FC_END */

/* 654 */ /* FC_UP */
/* 654 */ NdrFcShort( 0xfffffe4 ), /*
Offset= -28 (626) */
/* 656 */

```

```

/* FC_CARRAY */
/* 3 */
/* 658 */ NdrFcShort( 0x4 ), /* 4 */
/* 660 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */

/* 666 */ /* FC_PP */
/* FC_PAD */

/* 668 */ NdrFcShort( 0x4 ), /* 4 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x1 ), /* 1 */
/* 674 */ NdrFcShort( 0x0 ), /* 0 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ 0x12, 0x0, /* FC_UP */
/* 680 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (636) */
/* 682 */

/* 684 */ 0x5c, /* FC_END */
FC_PAD */

/* 686 */ /* FC_LONG */

FC_BOGUS_STRUCT */

/* 688 */ NdrFcShort( 0x8 ), /* 8 */
/* 690 */ NdrFcShort( 0x0 ), /* 0 */
/* 692 */ NdrFcShort( 0x6 ), /*
Offset= 6 (698) */
/* 694 */ 0x8, /*
FC_LONG */

/* 696 */ 0x5c, /* FC_POINTER */
FC_PAD */

/* 698 */ /* FC_END */

/* 700 */ /* FC_RP */
/* 700 */ NdrFcShort( 0xfffffd4 ), /*
Offset= -44 (656) */
/* 702 */

/* 704 */ NdrFcShort( 0x8 ), /* 8 */
/* 706 */ 0x1, /*
FC_BYTE */

/* 708 */ /* FC_END */

/* 710 */ /* FC_STRUCT */

```



```

/* 3 */ 0x3,
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /*
FC_LONG */

/* 714 */ 0x6, /* FC_SHORT */
/*
FC_SHORT */

/*
FC_EMBEDDED_COMPLEX */
/* 716 */ 0x0, /* 0 */

NdrFcShort( 0xfffff1 ), /*
Offset= -15 (702) */

/* 720 */ /* FC_END */

/*
FC_BOGUS_STRUCT */

/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0xa ), /*
Offset= 10 (736) */
/* 728 */ 0x8, /*
FC_LONG */

/* 730 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
/*
FC_EMBEDDED_COMPLEX */

/* 732 */ NdrFcShort( 0xfffffe8 ), /*
Offset= -24 (708) */
/* 734 */ 0x5c, /*
FC_PAD */

/* 736 */ /* FC_END */

0x0, /* FC_RP */
/* 738 */ NdrFcShort( 0xfffff0c ), /*
Offset= -244 (494) */
/* 740 */

/* FC_CARRAY */
/*
FC_CARRAY */

/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /*
FC_BYTE */

/* 750 */ /* FC_END */

/* FC_PSTRUCT */
/*
FC_PSTRUCT */

/* 752 */ NdrFcShort( 0x8 ), /* 8 */
/* 754 */

/* FC_PP */
/*
FC_PAD */

/* 756 */

```

```

/* FC_NO_REPEAT */
/* FC_PAD */
/* 758 */ NdrFcShort( 0x4 ), /* 4 */
/* 760 */ NdrFcShort( 0x4 ), /* 4 */
/* 762 */ 0x12, 0x0, /* FC_UP */
/* 764 */ NdrFcShort( 0xfffffe8 ), /*
Offset= -24 (740) */
/* 766 */

/* FC_END */

/* FC_LONG */
/*
FC_LONG */

/* 768 */ 0x8, /*
FC_LONG */

/* 770 */ /* FC_END */

/* FC_CARRAY */
/*
FC_CARRAY */

/* 772 */ NdrFcShort( 0x2 ), /* 2 */
/* 774 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ 0x6, /*
FC_SHORT */

/* 780 */ /* FC_END */

/* FC_PSTRUCT */
/*
FC_PSTRUCT */

/* 782 */ NdrFcShort( 0x8 ), /* 8 */
/* 784 */

/* FC_PP */
/*
FC_PAD */

/* 786 */

/* FC_NO_REPEAT */
/* FC_PAD */
/* 788 */ NdrFcShort( 0x4 ), /* 4 */
/* 790 */ NdrFcShort( 0x4 ), /* 4 */
/* 792 */ 0x12, 0x0, /* FC_UP */
/* 794 */ NdrFcShort( 0xfffffe8 ), /*
Offset= -24 (770) */
/* 796 */

/* FC_END */

/* FC_LONG */
/*
FC_LONG */

/* 798 */ 0x8, /*
FC_LONG */

/* 800 */ /* FC_END */

/* FC_CARRAY */
/*
FC_CARRAY */

/* 802 */ NdrFcShort( 0x4 ), /* 4 */
/* 804 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 806 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 808 */ 0x8, /*
FC_LONG */

/* 810 */ /* FC_END */

/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */

/* FC_PP */
/*
FC_PAD */

/* 816 */

/* FC_NO_REPEAT */
/* FC_PAD */
/* 818 */ NdrFcShort( 0x4 ), /* 4 */
/* 820 */ NdrFcShort( 0x4 ), /* 4 */
/* 822 */ 0x12, 0x0, /* FC_UP */
/* 824 */ NdrFcShort( 0xfffffe8 ), /*
Offset= -24 (800) */
/* 826 */

/* FC_END */

/* FC_LONG */
/*
FC_LONG */

/* 828 */ 0x8, /*
FC_LONG */

/* 830 */ /* FC_END */

/* FC_CARRAY */
/*
FC_CARRAY */

/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* 836 */ NdrFcShort( 0x0 ), /* 0 */
/* 838 */ 0xb, /*
FC_HYPER */

/* 840 */ /* FC_END */

/* FC_PSTRUCT */
/*
FC_PSTRUCT */

/* 842 */ NdrFcShort( 0x8 ), /* 8 */
/* 844 */

/* FC_PP */
/*
FC_PAD */

/* 846 */

/* FC_NO_REPEAT */
/* FC_PAD */
/* 848 */ NdrFcShort( 0x4 ), /* 4 */
/* 850 */ NdrFcShort( 0x4 ), /* 4 */
/* 852 */ 0x12, 0x0, /* FC_UP */
/* 854 */ NdrFcShort( 0xfffffe8 ), /*
Offset= -24 (830) */
/* 856 */

/* FC_END */

```

```

/* 858 */ 0x8,
FC_LONG */
/* 860 */
/* 862 */ NdrFcShort( 0x8 ), /* 8 */
/* 864 */ 0x8, /*
FC_LONG */
/* 866 */ 0x5c, /*
FC_PAD */
/* 868 */
/* 870 */ NdrFcShort( 0x8 ), /* 8 */
/* 872 */ 0x7, /* Corr
desc: FC_USHORT */
/* 874 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 876 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 878 */ NdrFcShort( 0xffffee ), /*
Offset= -18 (860) */
/* 880 */ 0x5c, /*
FC_PAD */
/* 882 */
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xffffee ), /*
Offset= -18 (868) */
/* 888 */ NdrFcShort( 0x0 ), /*
Offset= 0 (888) */
/* 890 */ 0x6, /*
FC_SHORT */
/* 892 */ 0x8, /*
FC_LONG */
/* 894 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 896 */ NdrFcShort( 0xffffd8 ), /*
Offset= -520 (376) */
/* 898 */ 0x5c, /*
FC_PAD */
/* 900 */
0x0, /* FC_UP */
/* 902 */ NdrFcShort( 0xffffef6 ), /*
Offset= -266 (636) */

```

```

/* 904 */ 0x12,
0x8, /* FC_UP [simple_pointer] */
/* 906 */ 0x1, /*
FC_BYTE */
/* 908 */
/* 910 */ 0x6, /*
FC_SHORT */
/* 912 */
/* 914 */ 0x8, /* FC_UP [simple_pointer] */
FC_LONG */
/* 916 */
/* 918 */ 0xb, /*
FC_HYPER */
/* 920 */
/* 922 */ 0xa, /*
FC_FLOAT */
/* 924 */
/* 926 */ 0xc, /*
FC_DOUBLE */
/* 928 */
/* 930 */ NdrFcShort( 0xffffd8c ), /*
Offset= -628 (302) */
/* 932 */
/* 934 */ NdrFcShort( 0xffffd8e ), /*
Offset= -626 (308) */
/* 936 */
/* 938 */ NdrFcShort( 0xffffda2 ), /*
Offset= -606 (332) */
/* 940 */
/* 942 */ NdrFcShort( 0xffffdb0 ), /*
Offset= -592 (350) */
/* 944 */
/* 946 */ NdrFcShort( 0xffffdbe ), /*
Offset= -578 (368) */
/* 948 */
/* 950 */ NdrFcShort( 0x2 ), /*
Offset= 2 (952) */
/* 952 */
0x0, /* FC_UP */

```

```

/* 954 */ NdrFcShort( 0x14 ), /*
Offset= -20 (974) */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /*
FC_SHORT */
/* 962 */ 0x1, /*
FC_BYTE */
/* 964 */ 0xb, /*
FC_HYPER */
/* 966 */
0x0, /* FC_UP */
/* 968 */ NdrFcShort( 0xfffff4 ), /*
Offset= -12 (956) */
/* 970 */
/* 972 */ 0x2, /*
FC_CHAR */
/* 974 */
FC_BOGUS_STRUCT */
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ), /* 0 */
/* 980 */ NdrFcShort( 0x0 ), /*
Offset= 0 (980) */
/* 982 */ 0x8, /*
FC_LONG */
/* 984 */ 0x6, /*
FC_SHORT */
/* 986 */ 0x6, /*
FC_SHORT */
/* 988 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
/* 990 */ NdrFcShort( 0xffffc28 ), /*
Offset= -984 (6) */
/* 992 */ 0x5c, /*
FC_PAD */
/* 994 */ 0xb4, /*
FC_USER_MARSHAL */
/* 996 */ NdrFcShort( 0x0 ), /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
/* 1002 */ NdrFcShort( 0xffffc18 ), /*
Offset= -1000 (2) */
/* 1004 */
0x4, /* FC_RP [allcled_on_stack] */

```

```

/* 1006 */ NdrFcShort( 0x6 ), /*
Offset= 6 (1012) */
/* 1008 */
0x0, /* FC_OP */
/* 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) */
0x0
};

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,
0x00,0x00,0x00,0x00,0x00,0x00}} */

/* 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={0xfEEEGAA2,0x84B1,0x11d2,{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,
};

static const MIDL_SERVER_INFO
ITPCC_ServerInfo =
{
&Object_StubDesc,
0,
_MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
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,
0,
NdrOleAllocate,
NdrOleFree,
0,
0,
0,
0,
0,
0,
0,
0,
0,
_MIDL_TypeFormatString.Format,
1, /* -error bounds_check flag */
0x20000, /* Ndr library version */
0,
0x600015b, /* MIDL Version 6.0.347 */
0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* cs routines */
0, /* proxy/server info */
0 /* Reserved5 */
};

const CInterfaceProxyVtbl *
_tpcCom_ps_ProxyVtblList[] =
{
{
(CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
0
};

```

```

const CInterfaceStubVtbl *
_tpcCom_ps_StubVtblList[] =
{
{
(CInterfaceStubVtbl *) &_ITPCCStubVtbl,
0
};
};

PCInterfaceName const
_tpcCom_ps_InterfaceNamesList[] =
{
"ITPCC",
0
};

#define _tpccComPs_CHECK_IID(n)
IID_GENERIC_CHECK_IID(
_tpcComPs, pIID, n)

int __stdcall _tpccComPs_IID_Lookup( const
IID * pIID, int * pIndex )
{
if(!_tpccComPs_CHECK_IID(0))
{
*pIndex = 0;
return 1;
}
return 0;
}

const ExtendedProxyFileInfo
tpccComPs_ProxyFileInfo =
{
(PCInterfaceProxyVtblList *) &
_tpcComPs_ProxyVtblList,
(PCInterfaceStubVtblList *) &
_tpcComPs_StubVtblList,
(const PCInterfaceName *) &
_tpcComPs_InterfaceNamesList,
0, // no delegation
&_tpccComPs_IID_Lookup,
1,
2,
0, /* table of [asyn_uid] interfaces */
0, /* Filler1 */
0, /* Filler2 */
0 /* Filler3 */
};

#ifdef !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\tpccComPs.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( )

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

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

    #include "rpcproxy.h"
    #ifndef __RPCPROXY_H_VERSION__
    #error this stub requires an updated version of
    <rpcproxy.h>
    #endif // __RPCPROXY_H_VERSION__

    #include "tpcc_com_ps.h"

    #define TYPE_FORMAT_STRING_SIZE 1003
    #define PROC_FORMAT_STRING_SIZE 253
    #define TRANSMIT_AS_TABLE_SIZE 0
    #define WIRE_MARSHAL_TABLE_SIZE 1

    typedef struct _MIDL_TYPE_FORMAT_STRING
    {
        short Pad;
        unsigned char Format[
        TYPE_FORMAT_STRING_SIZE ];
    } MIDL_TYPE_FORMAT_STRING;

    typedef struct _MIDL_PROC_FORMAT_STRING
    {
        short Pad;
        unsigned char Format[
        PROC_FORMAT_STRING_SIZE ];
    } MIDL_PROC_FORMAT_STRING;

    static RPC_SYNTAX_IDENTIFIER
    _RpcTransferSyntax =
    {{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,
    0x00,0x2B,0x10,0x48,0x60}}, {2,0}};

    extern const MIDL_TYPE_FORMAT_STRING
    __MIDL_TypeFormatString;
    extern const MIDL_PROC_FORMAT_STRING
    __MIDL_ProcFormatString;

    extern const MIDL_STUB_DESC
    Object_StubDesc;

    extern const MIDL_SERVER_INFO
    ITPCC_ServerInfo;
    extern const MIDL_STUBLESS_PROXY_INFO
    ITPCC_ProxyInfo;

    extern const
    USER_MARSHAL_ROUTINE_QUADRUPLE
    UserMarshalRoutines[
    WIRE_MARSHAL_TABLE_SIZE ];

    #if !defined(_RPC_WIN64_)

```

```

    #endif Invalid build platform for this stub.

    static const MIDL_PROC_FORMAT_STRING
    __MIDL_ProcFormatString =
    {
        0,
        {
            /* Procedure NewOrder */
            0x33,
            /* FC_AUTO_HANDLE */
            /*
            /* Old Flags: object,
            Oi2 */
            /* 2 */ NdrFcLong( 0x0 ), /* 0 */
            /* 6 */ NdrFcShort( 0x3 ), /* 3 */
            /* 8 */ NdrFcShort( 0x30 ), /* ia64
            Stack size/offset = 48 */
            /* 10 */ NdrFcShort( 0x0 ), /* 0 */
            /* 12 */ NdrFcShort( 0x8 ), /* 8 */
            /* 14 */ 0x47, /* Oi2
            Flags: srv must size, clt must size, has return,
            has ext, */
            0x3,
            /* 3 */
            /* 16 */ 0xa, /* 10 */
            0x7,
            /* Ext Flags: new
            corr desc, clt corr check, srv corr check, */
            /* 18 */ NdrFcShort( 0x20 ), /* 32 */
            /* 20 */ NdrFcShort( 0x20 ), /* 32 */
            /* 22 */ NdrFcShort( 0x0 ), /* 0 */
            /* 24 */ NdrFcShort( 0x0 ), /* 0 */
            /* Parameter txn_in */
            /* 26 */ NdrFcShort( 0x8b ), /* Flags:
            must size, must free, in, by val, */
            /* 28 */ NdrFcShort( 0x8 ), /* ia64
            Stack size/offset = 8 */
            /* 30 */ NdrFcShort( 0x3ce ), /* Type
            Offset=974 */
            /* Parameter txn_out */
            /* 32 */ NdrFcShort( 0x6113 ), /* Flags:
            must size, must free, out, simple ref, srv alloc
            size=24 */
            /* 34 */ NdrFcShort( 0x20 ), /* ia64
            Stack size/offset = 32 */
            /* 36 */ NdrFcShort( 0x3e0 ), /* Type
            Offset=992 */
            /* Return value */
            /* 38 */ NdrFcShort( 0x70 ), /* Flags:
            out, return, base type, */
            /* 40 */ NdrFcShort( 0x28 ), /* ia64
            Stack size/offset = 40 */
            /* 42 */ 0x8, /*
            FC_LONG */
            0x0,
            /* 0 */
            /* Procedure Payment */
            /* 44 */ 0x33, /*
            FC_AUTO_HANDLE */
            /* Old Flags: object,
            Oi2 */
            /* 46 */ NdrFcLong( 0x0 ), /* 0 */
            /* 50 */ NdrFcShort( 0x4 ), /* 4 */

```

```

            /* 52 */ NdrFcShort( 0x30 ), /* ia64
            Stack size/offset = 48 */
            /* 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 */

/* 0 */
/* Procedure StockLevel */

/* 132 */ 0x33, /*
FC_AUTO_HANDLE */

/* 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, */

/* 3 */
/* 148 */ 0xa, /* 10 */
/* 7,
/* 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 */

/* 0 */
/* 0 */

/* Procedure OrderStatus */

/* 176 */ 0x33, /*
FC_AUTO_HANDLE */

/* 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, */

/* 3 */
/* 192 */ 0xa, /* 10 */
/* 7,
/* 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 */

/* 0 */
/* 0 */

/* Procedure CallSetComplete */

/* 220 */ 0x33, /*
FC_AUTO_HANDLE */

/* Old Flags: object,
Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */

```

```

/* 228 */ NdrFcShort( 0x80 ), /* 8 */
Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2
Flags: has return, has ext, */

/* 1 */
/* 236 */ 0xa, /* 10 */
/* 1,
/* 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 */

/* 0 */
/* 0 */

}
};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */

/* 2 */
0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b6 ), /*
Offset= 950 (954) */
/* 6 */

/* 2b,
/*
FC_NON_ENCAPSULATED_UNION */
0x9,
/* FC_ULONG */
/* 8 */ 0x7, /* Corr
desc: FC_USHORT */

/* 10 */
/* 12 */ NdrFcShort( 0xff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 14 */ NdrFcShort( 0x2 ), /*
Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2f ), /* 47 */
/* 20 */ NdrFcLong( 0x14 ), /* 20 */
/* 24 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 26 */ NdrFcLong( 0x3 ), /* 3 */
/* 30 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 32 */ NdrFcLong( 0x11 ), /* 17 */
/* 36 */ NdrFcShort( 0x8001 ), /*
Simple arm type: FC_BYTE */
/* 38 */ NdrFcLong( 0x2 ), /* 2 */
/* 42 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 44 */ NdrFcLong( 0x4 ), /* 4 */

```

```

/* 48 */ NdrFcShort( 0x800a ), /*
Simple arm type: FC_FLOAT */
/* 50 */ NdrFcLong( 0x5 ), /* 5 */
/* 54 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 56 */ NdrFcLong( 0xb ), /* 11 */
/* 60 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_LONG */
/* 62 */ NdrFcLong( 0xa ), /* 10 */
/* 66 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 68 */ NdrFcLong( 0x6 ), /* 6 */
/* 72 */ NdrFcShort( 0xe8 ), /*
Offset= 232 (304) */
/* 74 */ NdrFcLong( 0x7 ), /* 7 */
/* 78 */ NdrFcShort( 0x800c ), /*
Simple arm type: FC_DOUBLE */
/* 80 */ NdrFcLong( 0x8 ), /* 8 */
/* 84 */ NdrFcShort( 0xe2 ), /*
Offset= 226 (310) */
/* 86 */ NdrFcLong( 0xd ), /* 13 */
/* 90 */ NdrFcShort( 0xf6 ), /*
Offset= 246 (336) */
/* 92 */ NdrFcLong( 0x9 ), /* 9 */
/* 96 */ NdrFcShort( 0x102 ), /*
Offset= 258 (354) */
/* 98 */ NdrFcLong( 0x2000 ), /* 8192
*/
/* 102 */ NdrFcShort( 0x10e ), /*
Offset= 270 (372) */
/* 104 */ NdrFcLong( 0x24 ), /* 36 */
/* 108 */ NdrFcShort( 0x304 ), /*
Offset= 772 (880) */
/* 110 */ NdrFcLong( 0x4024 ), /* 16420
*/
/* 114 */ NdrFcShort( 0x2fe ), /*
Offset= 766 (880) */
/* 116 */ NdrFcLong( 0x4011 ), /* 16401
*/
/* 120 */ NdrFcShort( 0x2fc ), /*
Offset= 764 (884) */
/* 122 */ NdrFcLong( 0x4002 ), /* 16386
*/
/* 126 */ NdrFcShort( 0x2fa ), /*
Offset= 762 (888) */
/* 128 */ NdrFcLong( 0x4003 ), /* 16387
*/
/* 132 */ NdrFcShort( 0x2f8 ), /*
Offset= 760 (892) */
/* 134 */ NdrFcLong( 0x4014 ), /* 16404
*/
/* 138 */ NdrFcShort( 0x2f6 ), /*
Offset= 758 (896) */
/* 140 */ NdrFcLong( 0x4004 ), /* 16388
*/
/* 144 */ NdrFcShort( 0x2f4 ), /*
Offset= 756 (900) */
/* 146 */ NdrFcLong( 0x4005 ), /* 16389
*/
/* 150 */ NdrFcShort( 0x2f2 ), /*
Offset= 754 (904) */
/* 152 */ NdrFcLong( 0x400b ), /* 16395
*/
/* 156 */ NdrFcShort( 0x2dc ), /*
Offset= 732 (888) */
/* 158 */ NdrFcLong( 0x400a ), /* 16394
*/
/* 162 */ NdrFcShort( 0x2da ), /*
Offset= 730 (892) */
/* 164 */ NdrFcLong( 0x4006 ), /* 16390
*/
/* 168 */ NdrFcShort( 0x2e4 ), /*
Offset= 740 (908) */
/* 170 */ NdrFcLong( 0x4007 ), /* 16391
*/

```

```

/* 174 */ NdrFcShort( 0x2da ), /*
Offset= 730 (904) */
/* 176 */ NdrFcLong( 0x4008 ), /* 16392
*/
/* 180 */ NdrFcShort( 0x2dc ), /*
Offset= 732 (912) */
/* 182 */ NdrFcLong( 0x400d ), /* 16397
*/
/* 186 */ NdrFcShort( 0x2da ), /*
Offset= 730 (916) */
/* 188 */ NdrFcLong( 0x4009 ), /* 16393
*/
/* 192 */ NdrFcShort( 0x2d8 ), /*
Offset= 728 (920) */
/* 194 */ NdrFcLong( 0x6000 ), /* 24576
*/
/* 198 */ NdrFcShort( 0x2d6 ), /*
Offset= 726 (924) */
/* 200 */ NdrFcLong( 0x400c ), /* 16396
*/
/* 204 */ NdrFcShort( 0x2d4 ), /*
Offset= 724 (928) */
/* 206 */ NdrFcLong( 0x10 ), /* 16 */
/* 210 */ NdrFcShort( 0x8002 ), /*
Simple arm type: FC_CHAR */
/* 212 */ NdrFcLong( 0x12 ), /* 18 */
/* 216 */ NdrFcShort( 0x8006 ), /*
Simple arm type: FC_SHORT */
/* 218 */ NdrFcLong( 0x13 ), /* 19 */
/* 222 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0x15 ), /* 21 */
/* 228 */ NdrFcShort( 0x800b ), /*
Simple arm type: FC_HYPER */
/* 230 */ NdrFcLong( 0x16 ), /* 22 */
/* 234 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 236 */ NdrFcLong( 0x17 ), /* 23 */
/* 240 */ NdrFcShort( 0x8008 ), /*
Simple arm type: FC_LONG */
/* 242 */ NdrFcLong( 0xe ), /* 14 */
/* 246 */ NdrFcShort( 0x2b2 ), /*
Offset= 690 (936) */
/* 248 */ NdrFcLong( 0x400e ), /* 16398
*/
/* 252 */ NdrFcShort( 0x2b6 ), /*
Offset= 694 (946) */
/* 254 */ NdrFcLong( 0x4010 ), /* 16400
*/
/* 258 */ NdrFcShort( 0x2b4 ), /*
Offset= 692 (950) */
/* 260 */ NdrFcLong( 0x4012 ), /* 16402
*/
/* 264 */ NdrFcShort( 0x270 ), /*
Offset= 624 (888) */
/* 266 */ NdrFcLong( 0x4013 ), /* 16403
*/
/* 270 */ NdrFcShort( 0x26e ), /*
Offset= 622 (892) */
/* 272 */ NdrFcLong( 0x4015 ), /* 16405
*/
/* 276 */ NdrFcShort( 0x26c ), /*
Offset= 620 (896) */
/* 278 */ NdrFcLong( 0x4016 ), /* 16406
*/
/* 282 */ NdrFcShort( 0x262 ), /*
Offset= 610 (892) */
/* 284 */ NdrFcLong( 0x4017 ), /* 16407
*/
/* 288 */ NdrFcShort( 0x25c ), /*
Offset= 604 (892) */
/* 290 */ NdrFcLong( 0x0 ), /* 0 */
/* 294 */ NdrFcShort( 0x0 ), /*
Offset= 0 (294) */
/* 296 */ NdrFcLong( 0x1 ), /* 1 */

```

```

/* 300 */ NdrFcShort( 0x0 ), /*
Offset= 0 (300) */
/* 302 */ NdrFcShort( 0xfffff ), /*
Offset= -1 (301) */
/* 304 */
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0xb, /*
FC_HYPER */
/* 310 */
/* 312 */ NdrFcShort( 0xe ), /*
Offset= 14 (326) */
/* 314 */
/* 316 */ NdrFcShort( 0x2 ), /* 2 */
/* 318 */ 0x9, /* Cor
desc: FC_ULONG */
/* 320 */ NdrFcShort( 0xfffc ), /* -4 */
/* 322 */ NdrFcShort( 0x1 ), /* Cor
flags: early, */
/* 324 */ 0x6, /*
FC_SHORT */
/* 326 */
/* 328 */ NdrFcShort( 0x8 ), /* 8 */
/* 330 */ NdrFcShort( 0xfffff0 ), /*
Offset= -16 (314) */
/* 332 */ 0x8, /*
FC_LONG */
/* 334 */ 0x5c, /*
FC_PAD */
/* 336 */
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x0 ), /* 0 */
/* 344 */ NdrFcShort( 0x0 ), /* 0 */
/* 346 */ 0xc0, /* 192
*/
/* 348 */ 0x0, /* 0 */
/* 350 */ 0x0, /* 0 */
/* 352 */ 0x0, /* 0 */
/* 354 */

```

```

0x2f,
/* FC_IP */
0x5a,
/* FC_CONSTANT_IID
*/
/* 356 */ NdrFcLong( 0x20400 ), /*
132096 */
/* 360 */ NdrFcShort( 0x0 ), /* 0 */
/* 362 */ NdrFcShort( 0x0 ), /* 0 */
/* 364 */ 0xc0, /* 192
*/
0x0,
/* 366 */ 0x0, /* 0
*/
/* 368 */ 0x0, /* 0
*/
/* 370 */ 0x0, /* 0
*/
/* 372 */ /* 70
*/
0x10, /* FC_UP [pointer_deref]
/* 374 */ NdrFcShort( 0x2 ), /*
Offset= 2 (376)
/* 376 */
0x12,
0x0, /* FC_UP
/* 378 */ NdrFcShort( 0x1e4 ), /*
Offset= 484 (862)
/* 380 */
*/
0x2a,
FC_ENCAPSULATED_UNION
/* 137
*/
/* 382 */ NdrFcShort( 0x20 ), /* 32
/* 384 */ NdrFcShort( 0xa ), /* 10
/* 386 */ NdrFcLong( 0x8 ), /* 8
/* 390 */ NdrFcShort( 0x50 ), /*
Offset= 80 (470)
/* 392 */ NdrFcLong( 0xd ), /* 13
/* 396 */ NdrFcShort( 0x70 ), /*
Offset= 112 (508)
/* 398 */ NdrFcLong( 0x9 ), /* 9
/* 402 */ NdrFcShort( 0x90 ), /*
Offset= 144 (546)
/* 404 */ NdrFcLong( 0xc ), /* 12
/* 408 */ NdrFcShort( 0xb0 ), /*
Offset= 176 (584)
/* 410 */ NdrFcLong( 0x24 ), /* 36
/* 414 */ NdrFcShort( 0x102 ), /*
Offset= 258 (672)
/* 416 */ NdrFcLong( 0x800d ), /* 32781
*/
/* 420 */ NdrFcShort( 0x11e ), /*
Offset= 286 (706)
/* 422 */ NdrFcLong( 0x10 ), /* 16
/* 426 */ NdrFcShort( 0x138 ), /*
Offset= 312 (738)
/* 428 */ NdrFcLong( 0x2 ), /* 2
/* 432 */ NdrFcShort( 0x14e ), /*
Offset= 334 (766)
/* 434 */ NdrFcLong( 0x3 ), /* 3
/* 438 */ NdrFcShort( 0x164 ), /*
Offset= 356 (794)
/* 440 */ NdrFcLong( 0x14 ), /* 20
/* 444 */ NdrFcShort( 0x17a ), /*
Offset= 378 (822)
/* 446 */ NdrFcShort( 0xfffff ), /*
Offset= -1 (445)
/* 448 */

```

```

0x21,
/* FC_BOGUS_ARRAY
0x3,
*/
/* 3 */
/* 450 */ NdrFcShort( 0x0 ), /* 0
/* 452 */ 0x19, /* Corr
desc: field pointer, FC_ULONG
*/
0x0,
/* 454 */ NdrFcShort( 0x0 ), /* 0
/* 456 */ NdrFcShort( 0x1 ), /* Corr
flags: early,
/* 458 */ NdrFcLong( 0xfffff ), /* -1
/* 462 */ NdrFcShort( 0x0 ), /* Corr
flags:
/* 464 */
0x12,
0x0, /* FC_UP
/* 466 */ NdrFcShort( 0xffff74 ), /*
Offset= -140 (326)
/* 468 */ 0x5c, /*
FC_PAD
*/
0x5b,
/* FC_END
*/
/* 470 */
0x1a,
*/
FC_BOGUS_STRUCT
0x3,
/* 3
*/
/* 472 */ NdrFcShort( 0x10 ), /* 16
/* 474 */ NdrFcShort( 0x0 ), /* 0
/* 476 */ NdrFcShort( 0x6 ), /*
Offset= 6 (482)
/* 478 */ 0x8, /*
FC_LONG
*/
0x40,
/* FC_STRUCTPAD4
*/
/* 480 */ 0x36, /*
FC_POINTER
*/
0x5b,
/* FC_END
*/
0x11,
0x0, /* FC_RP
/* 484 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (448)
/* 486 */
0x21,
/* FC_BOGUS_ARRAY
*/
0x3,
/* 3
*/
/* 488 */ NdrFcShort( 0x0 ), /* 0
/* 490 */ 0x19, /* Corr
desc: field pointer, FC_ULONG
*/
0x0,
/* 492 */ NdrFcShort( 0x0 ), /* 0
/* 494 */ NdrFcShort( 0x1 ), /* Corr
flags: early,
/* 496 */ NdrFcLong( 0xfffff ), /* -1
/* 500 */ NdrFcShort( 0x0 ), /* Corr
flags:
/* 502 */ 0x4c, /*
FC_EMBEDDED_COMPLEX
*/
0x0,
/* 504 */ NdrFcShort( 0xfffff58 ), /*
Offset= -168 (336)
/* 506 */ 0x5c, /*
FC_PAD
*/
0x5b,
/* FC_END
*/

```

```

/* 508 */
/*
FC_BOGUS_STRUCT
*/
0x3,
/* 3
*/
/* 510 */ NdrFcShort( 0x10 ), /* 16
/* 512 */ NdrFcShort( 0x0 ), /* 0
/* 514 */ NdrFcShort( 0x6 ), /*
Offset= 6 (520)
/* 516 */ 0x8, /*
FC_LONG
*/
0x40,
/* FC_STRUCTPAD4
*/
/* 518 */ 0x36, /*
FC_POINTER
*/
0x5b,
/* FC_END
*/
/* 520 */
0x11,
0x0, /* FC_RP
/* 522 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (486)
/* 524 */
0x21,
/* FC_BOGUS_ARRAY
*/
0x3,
/* 3
*/
/* 526 */ NdrFcShort( 0x0 ), /* 0
/* 528 */ 0x19, /* Corr
desc: field pointer, FC_ULONG
*/
0x0,
/* 530 */ NdrFcShort( 0x0 ), /* 0
/* 532 */ NdrFcShort( 0x1 ), /* Corr
flags: early,
/* 534 */ NdrFcLong( 0xfffff ), /* -1
/* 538 */ NdrFcShort( 0x0 ), /* Corr
flags:
/* 540 */ 0x4c, /*
FC_EMBEDDED_COMPLEX
*/
0x0,
/* 542 */ NdrFcShort( 0xffff44 ), /*
Offset= -188 (354)
/* 544 */ 0x5c, /*
FC_PAD
*/
0x5b,
/* FC_END
*/
/* 546 */
0x1a,
*/
FC_BOGUS_STRUCT
0x3,
/* 3
*/
/* 548 */ NdrFcShort( 0x10 ), /* 16
/* 550 */ NdrFcShort( 0x0 ), /* 0
/* 552 */ NdrFcShort( 0x6 ), /*
Offset= 6 (558)
/* 554 */ 0x8, /*
FC_LONG
*/
0x40,
/* FC_STRUCTPAD4
*/
/* 556 */ 0x36, /*
FC_POINTER
*/
0x5b,
/* FC_END
*/
/* 558 */
0x11,
0x0, /* FC_RP
/* 560 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (524)
/* 562 */

```

<pre> 0x21, /* FC_BOGUS_ARRAY */ /* 3 */ 0x3, /* 564 */ NdrFcShort(0x0), /* 0 */ /* 566 */ 0x19, /* Corr desc: field pointer, FC_ULONG */ 0x0, /* 568 */ NdrFcShort(0x0), /* 0 */ /* 570 */ NdrFcShort(0x1), /* Corr flags: early, */ /* 572 */ NdrFcLong(0xffffffff), /* -1 */ /* 576 */ NdrFcShort(0x0), /* Corr flags: */ /* 578 */ 0x12, 0x0, /* FC_UP */ /* 580 */ NdrFcShort(0x176), /* Offset= 374 (954) */ /* 582 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 584 */ 0x1a, FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 586 */ NdrFcShort(0x10), /* 16 */ /* 588 */ NdrFcShort(0x0), /* 0 */ /* 590 */ NdrFcShort(0x6), /* Offset= 6 (596) */ /* 592 */ 0x8, /* FC_LONG */ 0x40, /* FC_STRUCTPAD4 */ /* 594 */ 0x36, /* FC_POINTER */ 0x5b, /* FC_END */ /* 596 */ 0x11, 0x0, /* FC_RP */ /* 598 */ NdrFcShort(0xfffffdc), /* Offset= -36 (562) */ /* 600 */ 0x2f, /* FC_IP */ 0x5a, /* FC_CONSTANT_IID */ /* 602 */ NdrFcLong(0x2f), /* 47 */ /* 606 */ NdrFcShort(0x0), /* 0 */ /* 608 */ NdrFcShort(0x0), /* 0 */ /* 610 */ 0xc0, /* 192 */ 0x0, /* 612 */ 0x0, /* 0 */ 0x0, /* 614 */ 0x0, /* 0 */ 0x0, /* 616 */ 0x0, /* 0 */ 0x46, /* 618 */ /* 70 */ 0x1b, /* FC_CARRAY */ </pre>	<pre> /* 620 */ NdrFcShort(0x10), /* /* 622 */ 0x19, /* desc: field pointer, FC_ULONG */ 0x0, /* 3 */ /* 624 */ NdrFcShort(0x4), /* 4 */ /* 626 */ NdrFcShort(0x1), /* Corr flags: early, */ /* 628 */ 0x1, /* FC_BYTE */ 0x5b, /* FC_END */ /* 630 */ 0x1a, FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 632 */ NdrFcShort(0x18), /* 24 */ /* 634 */ NdrFcShort(0x0), /* 0 */ /* 636 */ NdrFcShort(0xa), /* Offset= 10 (646) */ /* 638 */ 0x8, /* FC_LONG */ 0x8, /* FC_LONG */ /* 640 */ 0x4c, /* FC_EMBEDDED_COMPLEX */ 0x0, /* 0 */ /* 642 */ NdrFcShort(0xffffffff6), /* Offset= -42 (600) */ /* 644 */ 0x36, /* FC_POINTER */ 0x5b, /* FC_END */ /* 646 */ 0x12, 0x0, /* FC_UP */ /* 648 */ NdrFcShort(0xfffffe2), /* Offset= -30 (618) */ /* 650 */ 0x21, /* FC_BOGUS_ARRAY */ 0x3, /* 3 */ /* 652 */ NdrFcShort(0x0), /* 0 */ /* 654 */ 0x19, /* Corr desc: field pointer, FC_ULONG */ 0x0, /* 0 */ /* 656 */ NdrFcShort(0x0), /* 0 */ /* 658 */ NdrFcShort(0x1), /* Corr flags: early, */ /* 660 */ NdrFcLong(0xffffffff), /* -1 */ /* 664 */ NdrFcShort(0x0), /* Corr flags: */ /* 666 */ 0x12, 0x0, /* FC_UP */ /* 668 */ NdrFcShort(0xfffffda), /* Offset= -38 (630) */ /* 670 */ 0x5c, /* FC_PAD */ 0x5b, /* FC_END */ /* 672 */ 0x1a, FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 674 */ NdrFcShort(0x10), /* 16 */ </pre>	<pre> /* 676 */ NdrFcShort(0x6), /* 0 */ Offset= 6 (684) */ /* 680 */ 0x8, /* FC_LONG */ 0x40, /* FC_STRUCTPAD4 */ /* 682 */ 0x36, /* FC_POINTER */ 0x5b, /* FC_END */ /* 684 */ 0x11, 0x0, /* FC_RP */ /* 686 */ NdrFcShort(0xfffffdc), /* Offset= -36 (650) */ /* 688 */ 0x1d, /* FC_SMFARRAY */ 0x0, /* 0 */ /* 690 */ NdrFcShort(0x8), /* 8 */ /* 692 */ 0x1, /* FC_BYTE */ 0x5b, /* FC_END */ /* 694 */ 0x15, /* FC_STRUCT */ 0x3, /* 3 */ /* 696 */ NdrFcShort(0x10), /* 16 */ /* 698 */ 0x8, /* FC_LONG */ 0x6, /* FC_SHORT */ /* 700 */ 0x6, /* FC_SHORT */ 0x4c, FC_EMBEDDED_COMPLEX */ /* 702 */ 0x0, /* 0 */ NdrFcShort(0xfffff1), /* Offset= -15 (688) */ 0x5b, /* FC_END */ /* 706 */ 0x1a, FC_BOGUS_STRUCT */ 0x3, /* 3 */ /* 708 */ NdrFcShort(0x20), /* 32 */ /* 710 */ NdrFcShort(0x0), /* 0 */ /* 712 */ NdrFcShort(0xa), /* Offset= 10 (722) */ /* 714 */ 0x8, /* FC_LONG */ 0x40, /* FC_STRUCTPAD4 */ /* 716 */ 0x36, /* FC_POINTER */ 0x4c, FC_EMBEDDED_COMPLEX */ /* 718 */ 0x0, /* 0 */ NdrFcShort(0xfffffe7), /* Offset= -25 (694) */ 0x5b, /* FC_END */ /* 722 */ </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------


```

0x0, /* FC_RP */ 0x11,
/* 724 */ NdrFcShort( 0xffff12 ), /*
Offset= -238 (486) */
/* 726 */

/* FC_CARRAY */
0x1b,
0x0,
/* 0 */
/* 728 */ NdrFcShort( 0x1 ), /* 1 */
/* 730 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* */
0x0,
/* 732 */ NdrFcShort( 0x0 ), /* 0 */
/* 734 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 736 */ 0x1, /*
FC_BYTE */

/* FC_END */
0x5b,
/* 738 */

0x1a,
/*
FC_BOGUS_STRUCT */

/* 3 */
0x3,
/* 740 */ NdrFcShort( 0x10 ), /* 16 */
/* 742 */ NdrFcShort( 0x0 ), /* 0 */
/* 744 */ NdrFcShort( 0x6 ), /*
Offset= 6 (750) */
/* 746 */ 0x8, /*
FC_LONG */

/* FC_STRUCTPAD4 */
0x40,
/*
/* 748 */ 0x36, /*
FC_POINTER */

/* FC_END */
0x5b,
/* 750 */

0x12,
0x0, /* FC_UP */
/* 752 */ NdrFcShort( 0xfffffe6 ), /*
Offset= -26 (726) */
/* 754 */

/* FC_CARRAY */
0x1b,
0x1,
/* 1 */
/* 756 */ NdrFcShort( 0x2 ), /* 2 */
/* 758 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* */
0x0,
/* 760 */ NdrFcShort( 0x0 ), /* 0 */
/* 762 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 764 */ 0x6, /*
FC_SHORT */

/* FC_END */
0x5b,
/* 766 */

0x1a,
/*
FC_BOGUS_STRUCT */

/* 3 */
0x3,
/* 768 */ NdrFcShort( 0x10 ), /* 16 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x6 ), /*
Offset= 6 (778) */
/* 774 */ 0x8, /*
FC_LONG */

0x40, /* FC_STRUCTPAD4
/* 776 */ 0x36, /*
FC_POINTER */

/* FC_END */
0x5b,
/* 778 */

0x12,
0x0, /* FC_UP */
/* 780 */ NdrFcShort( 0xfffffe6 ), /*
Offset= -26 (754) */
/* 782 */

/* FC_CARRAY */
0x1b,
0x3,
/* 3 */
/* 784 */ NdrFcShort( 0x4 ), /* 4 */
/* 786 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* */
0x0,
/* 788 */ NdrFcShort( 0x0 ), /* 0 */
/* 790 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 792 */ 0x8, /*
FC_LONG */

/* FC_END */
0x5b,
/* 794 */

0x1a,
/*
FC_BOGUS_STRUCT */

/* 3 */
0x3,
/* 796 */ NdrFcShort( 0x10 ), /* 16 */
/* 798 */ NdrFcShort( 0x0 ), /* 0 */
/* 800 */ NdrFcShort( 0x6 ), /*
Offset= 6 (806) */
/* 802 */ 0x8, /*
FC_LONG */

/* FC_STRUCTPAD4 */
0x40,
/*
/* 804 */ 0x36, /*
FC_POINTER */

/* FC_END */
0x5b,
/* 806 */

0x12,
0x0, /* FC_UP */
/* 808 */ NdrFcShort( 0xfffffe6 ), /*
Offset= -26 (782) */
/* 810 */

/* FC_CARRAY */
0x1b,
0x7,
/* 7 */
/* 812 */ NdrFcShort( 0x8 ), /* 8 */
/* 814 */ 0x19, /* Corr
desc: field pointer, FC_ULONG */

/* */
0x0,
/* 816 */ NdrFcShort( 0x0 ), /* 0 */
/* 818 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 820 */ 0xb, /*
FC_HYPER */

/* FC_END */
0x5b,
/* 822 */

0x1a,
/*
FC_BOGUS_STRUCT */

/* 3 */
0x3,
/* 824 */ NdrFcShort( 0x90 ), /* 96 */
/* 828 */ NdrFcShort( 0x6 ), /*
Offset= 6 (834) */
/* 830 */ 0x8, /*
FC_LONG */

/* FC_STRUCTPAD4 */
0x40,
/*
/* 832 */ 0x36, /*
FC_POINTER */

/* FC_END */
0x5b,
/* 834 */

0x12,
0x0, /* FC_UP */
/* 836 */ NdrFcShort( 0xfffffe6 ), /*
Offset= -26 (810) */
/* 838 */

/* FC_UP */
0x15,
/* FC_STRUCT */
0x3,
/* 3 */
/* 840 */ NdrFcShort( 0x8 ), /* 8 */
/* 842 */ 0x8, /*
FC_LONG */

/* FC_LONG */
0x8,
/* 844 */ 0x5c, /*
FC_PAD */

/* FC_END */
0x5b,
/* 846 */

0x1b,
/* FC_CARRAY */
0x3,
/* 3 */
/* 848 */ NdrFcShort( 0x8 ), /* 8 */
/* 850 */ 0x7, /* Corr
desc: FC_USHORT */

0x0,
/* */
/* 852 */ NdrFcShort( 0xffc8 ), /* -56 */
/* 854 */ NdrFcShort( 0x1 ), /* Corr
flags: early, */
/* 856 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */

0x0,
/* 0 */
/* 858 */ NdrFcShort( 0xfffffec ), /*
Offset= -20 (838) */
/* 860 */ 0x5c, /*
FC_PAD */

0x5b,
/* FC_END */
/* 862 */

0x1a,
/*
FC_BOGUS_STRUCT */

/* 3 */
0x3,
/* 864 */ NdrFcShort( 0x38 ), /* 56 */
/* 866 */ NdrFcShort( 0xfffffec ), /*
Offset= -20 (846) */
/* 868 */ NdrFcShort( 0x0 ), /*
Offset= 0 (868) */
/* 870 */ 0x6, /*
FC_SHORT */

0x6,
/* FC_SHORT */
/* 872 */ 0x8, /*
FC_LONG */

0x8,
/* FC_LONG */
/* 874 */ 0x40, /*
FC_STRUCTPAD4 */

```

```

/* 0x4c,
FC_EMBEDDED_COMPLEX */
/* 876 */ 0x0, /* 0 */

NdrFcShort( 0xffffe0f ), /*
Offset= -497 (380) */

/* 880 */
/* FC_END */
0x5b,
0x12,
0x0, /* FC_UP */
/* 882 */ NdrFcShort( 0xfffff04 ), /*
Offset= -252 (630) */
/* 884 */

0x8, /* FC_UP [simple_pointer] */
/* 886 */ 0x1, /*
FC_BYTE */

0x5c,
/* FC_PAD */

/* 888 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 890 */ 0x6, /*
FC_SHORT */

0x5c,
/* FC_PAD */

/* 892 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 894 */ 0x8, /*
FC_LONG */

0x5c,
/* FC_PAD */

/* 896 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 898 */ 0xb, /*
FC_HYPER */

0x5c,
/* FC_PAD */

/* 900 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 902 */ 0xa, /*
FC_FLOAT */

0x5c,
/* FC_PAD */

/* 904 */
0x12,
0x8, /* FC_UP [simple_pointer] */
/* 906 */ 0xc, /*
FC_DOUBLE */

0x5c,
/* FC_PAD */

/* 908 */
0x0, /* FC_UP */
/* 910 */ NdrFcShort( 0xffffda2 ), /*
Offset= -606 (304) */
/* 912 */

0x12,
0x10, /* FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xffffda4 ), /*
Offset= -604 (310) */
/* 916 */

0x12,
0x10, /* FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xffffdba ), /*
Offset= -582 (336) */
/* 920 */

0x12,
0x10, /* FC_UP [pointer_deref] */

```

```

/* 922 */ NdrFcShort( 0xffffdc8 ), /*
Offset= -568 (354) */
0x12,
0x10, /* FC_UP [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( 0xffffff4 ), /*
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 */

0x6,
/* FC_SHORT */

/* 968 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */

0x0,
/* 0 */

/* 970 */ NdrFcShort( 0xffffc3c ), /*
Offset= -964 (6) */

```

```

/* 972 */ 0x5c, /*
FC_PAD */
/* 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,
/* FC_RP [allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /*
Offset= 6 (992) */
/* 988 */

0x13,
/* FC_OP */
/* 990 */ NdrFcShort( 0xfffffdc ), /*
Offset= -36 (954) */
/* 992 */ 0xb4, /*
FC_USER_MARSHAL */

0x83,
/* 131 */

/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xffffff4 ), /*
Offset= -12 (988) */

0x0
}
};

static const
USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[
WIRE_MARSHAL_TABLE_SIZE ] =
{
{
VARIANT_UserSize
,VARIANT_UserMarshal
,VARIANT_UserUnmarshal
,VARIANT_UserFree
}
};

/* Standard interface:
__MIDL_itf_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,
0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEEA2,0x84B1,0x11d2,{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,
    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,
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    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 */
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* cs routines */
0, /* proxy/server info */
0 /* Reserved5 */
};

const CInterfaceProxyVtbl *
_tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl *
_tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const
_tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID(
_tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const
IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo
tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &
_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &
_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

#endif /* defined(_M_IA64) ||
defined(_M_AMD64)*/

```

Appendix B Database Load

SetupOldLoader.cmd

```
@echo off

set DoDBCREATE=TRUE
set DoBULKINSERT=TRUE
set DoCLEANUP=TRUE
set DoBACKUP=FALSE
set MaxDOP=24

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=20
set WAREHOUSES=68000
set DB=tpcc
set PASSWD=ssdl
set SERVER=sqlwestmere

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

if %PROCESSOR_ARCHITECTURE% == x86 (set
EXECUTABLEPATH=%HOMEPATH%\Executables\x86
) else set
EXECUTABLEPATH=%HOMEPATH%\Executables\x64

%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
%HOMEDRIVE%%OUTPUTDIR%\%OUTPUTNUMBER%
R% 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
```

```
echo Starting database creation
time /t >> %OUTPUTPATH%\time.txt
echo Starting database creation >>
%OUTPUTPATH%\time.txt
%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E -i CreateDatabase.sql -e -o
%OUTPUTPATH%\CreateDatabase.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT
if '%1' == 'DBCREATE'
%EXECUTABLEPATH%\Semaphore -release
echo DBCREATE done %OUTPUTPATH%\time.txt
time /t %OUTPUTPATH%\time.txt
if '%1' == 'DBCREATE' goto :EOF

:BULKINSERT
if NOT '%DoBULKINSERT%' == 'TRUE' goto
:CLEANUP
rem
rem BULKINSERT loads each table through a set of
concurrent bulk inserts
rem After each table is loaded, the table is indexed
rem

echo Starting Bulk Inserts
%HOMEDRIVE%
cd %HOMEDIR%\TablesAndIndexes

osql -E -d%DB% -
i%HOMEPATH%\Database\dbopt1.sql -e -o
%OUTPUTPATH%\dbopt1.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'show advanced options',1" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'max degree',%MaxDOP%" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT

rem go :AROUND
echo Dropping and Re-Creating Tables
osql -E -d%DB% -i>CreateTables.sql -e -o
%OUTPUTPATH%\CreateTables.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT

rem Special case for Item, since it is only one file
echo Starting create indexes for Item
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreateITEMIn
dexes.sql -e -
o%OUTPUTPATH%\NewCreateItemIndexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
echo Starting bulk inserts for Item
%EXECUTABLEPATH%\stockldr\tpccldr -s1 -W10 -
titem -Pssdl -i1 -o1 -f%OUTPUTPATH%\loaditem.out -
L%OUTPUTPATH%\
if ERRORLEVEL 1 goto :ERROR_EXIT

rem :AROUND
echo Starting bulk inserts of item warehouse customer
orders

for %%i in (warehouse customer orders) do call
:BULKINSERTi %%i
rem for %%i in (customer orders) do call
:BULKINSERTi %%i
if ERRORLEVEL 1 goto :ERROR_EXIT

goto :CLEANUP

:BULKINSERTi
echo Starting bulk inserts for %1
```

```
for /I %%j in (1,1,%THREADS%) do start cmd /C
%HOMEPATH%\SetupOldLoader BULKINSERTn %1
%EXECUTABLEPATH%\Semaphore -wait
%THREADS%

echo Starting create indexes for %1
osql -E -d%DB% -
i%HOMEPATH%\TablesAndIndexes\NewCreate%1Ind
exes.sql -e -o
%OUTPUTPATH%\NewCreate%1Indexes.out
if ERRORLEVEL 1 goto :ERROR_EXIT
goto :EOF

:BULKINSERTn
set /a REM=WAREHOUSES %% THREADS
set /a WAREHOUSEPERTHREAD=WAREHOUSES /
THREADS
set /a TEMP=%3-1
set /a
STARTWAREHOUSE=%TEMP%*%WAREHOUSEPERTH
READ%
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%+%WAREH
OUSEPERTHREAD%-1
md %OUTPUTPATH%\%3
%EXECUTABLEPATH%\stockldr\tpccldr -t%2 -Pssdl -
s%STARTWAREHOUSE% -W%ENDWAREHOUSE% -i1
-o1 -
f%OUTPUTPATH%\load%2%STARTWAREHOUSE%.ou
t -L%OUTPUTPATH%\%3\ -S%SERVER%

%EXECUTABLEPATH%\Semaphore -release
goto :EOF

:CLEANUP
if NOT '%DoCLEANUP%' == 'TRUE' goto :BACKUP
rem
rem CLEANUP sets statistics and lock options
rem

echo Setting Cleanup Options

%HOMEDRIVE%
cd %HOMEDIR%\Database
osql -E -d%DB% -
i%HOMEPATH%\Database\dbopt2.sql -e -o
%OUTPUTPATH%\dbopt2.out
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"sp_configure 'max degree',1" -b
if ERRORLEVEL 1 goto :ERROR_EXIT
osql -E -Q"reconfigure with override" -b
if ERRORLEVEL 1 goto :ERROR_EXIT

cd %HOMEDIR%\StoredProcedures
for %%i in (Create*Proc.sql) do osql -E -d%DB% -
i%HOMEPATH%\StoredProcedures\%%i -e -o
%OUTPUTPATH%\%%i.out -b

osql -E -Q"sp_dboption %DB%,trunc,FALSE" -b
>%OUTPUTPATH%\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 -e -o
%OUTPUTPATH%\Backup.out -b
if ERRORLEVEL 1 goto :ERROR_EXIT

:DONE

echo Done! Check for output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%

goto :EOF

:NO_HOMEDIR
echo directory %HOMEDRIVE%\%HOMEDIR% does
not exist
goto :ERROR_EXIT

:NO_OUTPUTDIR
echo directory %OutPUTDRIVE%\%OUTPUTDIR%
does not exist
goto :ERROR_EXIT

:ERROR_EXIT
echo SetupOldLoader aborted due to errors
echo Check output in %OUTPUTPATH%
%HOMEDRIVE%
cd %HOMEDIR%
exit /B

```

tpcc.h

```

// File: TPCC.H
//
// Microsoft TPC-C Kit Ver. 4.51
//
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2003, 2005
// Purpose: Header file for TPC-C
database loader

// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.51"

// General headers
#include <windows.h>
#include <winbase.h>
#include <stdlib.h>
#include <stdio.h>
#include <process.h>
#include <stddef.h>
#include <stdarg.h>
#include <string.h>
#include <time.h>
#include <sys/timeb.h>
#include <sys/types.h>

```

```

#include <math.h>
// ODBC headers
#include <sql.h>
#include <sqlext.h>
#include <odbcss.h>

// General constants
#define MILLI 1000
#define FALSE 0
#define TRUE 1
#define UNDEF -1

#define MINPRINTASCII 32
#define MAXPRINTASCII 126

// Default environment constants
#define SERVER ""
#define DATABASE "tpcc"
#define USER "sa"
#define PASSWORD ""

// Default loader arguments
#define BATCH 10000

#define DEFLDPACKSIZE 32768
#define LOADER_RES_FILE "\\load.out"
#define LOADER_LOG_PATH "\\."
#define LOADER_NURAND_C 123
#define DEF_STARTING_WAREHOUSE 1
#define BUILD_INDEX 1 // build both
data and indexes
#define INDEX_ORDER 1 // build
indexes before load
#define SCALE_DOWN 0 // build a normal scale
database
#define INDEX_SCRIPT_PATH "scripts"

typedef struct
{
char *server;
char *database;
char *user;
char *password;
BOOL tables_all;
// set if loading all tables
BOOL table_item;
// set if loading ITEM table specifically
BOOL table_warehouse; // set if
loading WAREHOUSE, DISTRICT, and STOCK

```

```

BOOL table_customer;
// set if loading CUSTOMER and
HISTORY BOOL table_orders; // set if
loading NEW-ORDER, ORDERS, ORDER-LINE
long num_warehouses;
long batch;
long verbose;
long pack_size;
char *loader_res_file;
char *log_path;
char *synch_servername;
long case_sensitivity;
long starting_warehouse;
long build_index;
long index_order;
long scale_down;
char *index_script_path;
} TPCCCLR_ARGS;

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50
#define D_NAME_LEN 10
#define FIRST_NAME_LEN 16
#define MIDDLE_NAME_LEN 2
#define PHONE_LEN 16
#define CREDIT_LEN 2
#define C_DATA_LEN 500
#define H_DATA_LEN 24
#define DIST_INFO_LEN 24
#define MAX_OL_NEW_ORDER_ITEMS 15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN 25
#define OL_DIST_INFO_LEN 24
#define C_SINCE_LEN 23
#define H_DATE_LEN 23
#define OL_DELIVERY_D_LEN 23
#define O_ENTRY_D_LEN 23

// Functions in random.c

```

```

void seed();
long irand();
double drand();
void WUCreate();
short WURand();
long RandomNumber(long lower, long upper);

// Functions in getargs.c;
void GetArgsLoader();
void GetArgsLoaderUsage();

// Functions in time.c
long TimeNow();

// Functions in strings.c
void MakeAddress();
void LastName();
int MakeAlphaString();
int MakeAlphaStringPadded();
int MakeOriginalAlphaString();
int MakeNumberString();
int MakeZipNumberString();
void InitString();
void InitAddress();
void PaddString();

```

tpccldr.c

```

//=====
// File: TPCCLDR.C
// Microsoft TPC-C Kit Ver. 4.51
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001, 2002, 2003
// Purpose: Source file for TPC-C
database loader
//=====
// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS 100000
#define MAXITEMS_SCALE_DOWN 100
#define CUSTOMERS_PER_DISTRICT 3000
#define CUSTOMERS_SCALE_DOWN 30
#define 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
#define MAX_SQL_ERRORS 10

// Functions declarations
void HandleErrorDBC (SQLHDBC hdbc1);
long NURand();
void LoadItem();
void LoadWarehouse();
void Stock();

```

```

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
    double ol;
    long ol_i_id;
    long ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long o_id;
    short o_d_id;
    long o_w_id;
    long o_c_id;
    short o_carrier_id;
    short o_ol_cnt;
    short o_all_local;
    ORDER_LINE_STRUCT o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long c_id;
    short c_d_id;
    long c_w_id;
    char c_first[FIRST_NAME_LEN+1];
    char c_middle[MIDDLE_NAME_LEN+1];
    char c_last[LAST_NAME_LEN+1];
    char c_street_1[ADDRESS_LEN+1];
    char c_street_2[ADDRESS_LEN+1];
    char c_city[ADDRESS_LEN+1];
    char c_state[STATE_LEN+1];
    char c_zip[ZIP_LEN+1];
    char c_phone[PHONE_LEN+1];
    char c_credit[CREDIT_LEN+1];
} CUSTOMER_STRUCT;

```

```

double c_credit_lim;
double c_discount;
char c_balance[6];
double c_ytd_payment;
short c_payment_cnt;
short c_delivery_cnt;
char c_data[C_DATA_LEN+1];
double h_amount;
char h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char c_last[LAST_NAME_LEN+1];
    char c_first[FIRST_NAME_LEN+1];
    long c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long time_start;
} LOADER_TIME_STRUCT;

// Global variables
char szLastError[300];
HENV henv;
HDBC hdbc;
// for SQL
Server version verification
HDBC i_hdbc1;
// for ITEM table
HDBC w_hdbc1;
// for WAREHOUSE,
DISTRICT, STOCK
HDBC c_hdbc1;
// for CUSTOMER
HDBC c_hdbc2;
// for HISTORY
HDBC o_hdbc1;
// for ORDERS
HDBC o_hdbc2;
// for NEW-ORDER
HDBC o_hdbc3;
// for ORDER-LINE
HSTMT v_hstmt;
// for SQL Server version
verification
HSTMT i_hstmt1;
HSTMT w_hstmt1;
HSTMT c_hstmt1, c_hstmt2;
HSTMT o_hstmt1, o_hstmt2, o_hstmt3;
long total_db_errors = 0;
ORDERS_STRUCT orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];

```

```

long      orders_rows_loaded;
double    new_order_rows_loaded;
double    order_line_rows_loaded;
long      history_rows_loaded;
long      customer_rows_loaded;
double    stock_rows_loaded;
long      district_rows_loaded;
long      item_rows_loaded;
long      warehouse_rows_loaded;
long      main_time_start;
long      main_time_end;
long      max_items;
long      customers_per_district;
long      orders_per_district;
long      first_new_order;
long      last_new_order;

TPCCLDR_ARGS  *aptr, argc;

//=====
//
// Function name: main
//
//=====
int main(int argc, char **argv)
{
    DWORD
dwThreadId[MAX_MAIN_THREADS];
HANDLE
hThread[MAX_MAIN_THREADS];
FILE      *fLoader;
char      buffer[255];
int
i;

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

    printf("\n*****\n");
    printf("\n*");
    printf("\n* Microsoft SQL Server");
    printf("\n*");
    printf("\n* TPC-C BENCHMARK KIT:");
    printf("\n* Database loader");
    printf("\n* TPCKIT_VER);");
    printf("\n*");
    printf("\n*****\n");
}

```

```

// process command line arguments
aptr = &argc;
GetArgsLoader(argc, argv, aptr);

    printf("Build interface is ODBC.\n");

    if (aptr->build_index == 0)
        printf("Data load only - no
index creation.\n");
    else
        {
            printf("Data load and index
creation.\n");
            if (aptr->index_order == 0)

                printf("Clustered indexes will be created
after bulk load.\n");
            else

                printf("Clustered indexes will be created
before bulk load.\n");
        }

    if (aptr->index_order == 0)
        printf("Clustered indexes
will be created after bulk load.\n");
    else
        printf("Clustered indexes
will be created before bulk load.\n");

    // set database scale values
    if (aptr->scale_down == 1)
        {
            printf("*** Scaled Down
Database ***\n");
            max_items =
MAXITEMS_SCALE_DOWN;
            customers_per_district =
CUSTOMERS_SCALE_DOWN;
            orders_per_district =
ORDERS_SCALE_DOWN;
            first_new_order = 0;
            last_new_order = 30;
        }
    else
        {
            max_items = MAXITEMS;
            customers_per_district =
CUSTOMERS_PER_DISTRICT;
            orders_per_district =
ORDERS_PER_DISTRICT;
            first_new_order = 2100;
            last_new_order = 3000;
        }

    // open connections to SQL Server
    OpenConnections();

    // open file for loader results
    fLoader = fopen(aptr->loader_res_file,
"w");

    if (fLoader == NULL)
        {
            printf("Error, loader result
file open failed.");
            exit(-1);
        }

    // start loading data
    sprintf(buffer,"TPC-C load started for %ld
warehouses. starting warehouse =%ld\n", (aptr-
>num_warehouses-aptr->starting_warehouse+1),

```

```

aptr-
>starting_warehouse);
{
    printf(buffer,"SCALED
DOWN DATABASE.\n");
}

    printf("%s",buffer);
    fprintf(fLoader,"%s",buffer);

    main_time_start = (TimeNow() / MILLI);

    // start parallel load threads
    if (aptr->tables_all || aptr->table_item)
        {
            fprintf(fLoader, "\nStarting
loader threads for: item\n");

            hThread[0] =
CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadItem,
NULL,
0,
&dwThreadId[0]);

            if (hThread[0] == NULL)
                {
                    printf("Error,
failed in creating creating thread = 0.\n");
                    exit(-1);
                }

            if (aptr->tables_all || aptr-
>table_warehouse)
                {
                    fprintf(fLoader, "Starting
loader threads for: warehouse\n");

                    hThread[1] =
CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
NULL,
0,
&dwThreadId[1]);

                    if (hThread[1] == NULL)
                        {
                            printf("Error,
failed in creating creating thread = 1.\n");
                            exit(-1);
                        }
                }
        }
}

```

```

    }
    if (aptr->tables_all || aptr->table_customer)
    {
        fprintf(fLoader, "Starting loader threads for: customer\n");
        hThread[2] =
        CreateThread(NULL,
        0,
        (LPTHREAD_START_ROUTINE) LoadCustomer,
        NULL,
        0,
        &dwThreadID[2]);
        if (hThread[2] == NULL)
        {
            printf("Error, failed in creating creating main thread = 2.\n");
            exit(-1);
        }
        if (aptr->tables_all || aptr->table_orders)
        {
            fprintf(fLoader, "Starting loader threads for: orders\n");
            hThread[3] =
            CreateThread(NULL,
            0,
            (LPTHREAD_START_ROUTINE) LoadOrders,
            NULL,
            0,
            &dwThreadID[3]);
            if (hThread[3] == NULL)
            {
                printf("Error, failed in creating creating main thread = 3.\n");
                exit(-1);
            }
            // Wait for threads to finish...
            for (i=0; i<MAX_MAIN_THREADS; i++)
            {
                if (hThread[i] != NULL)
                {
                    WaitForSingleObject( hThread[i],
                    INFINITE );
                    CloseHandle(hThread[i]);
                }
            }
        }
    }
}

```

```

    hThread[i] =
    NULL;
    }
    main_time_end = (TimeNow() / MILLI);
    sprintf(buffer, "\nTPC-C load completed successfully in %ld minutes.\n",
    (main_time_end - main_time_start)/60);
    printf("%s", buffer);
    fprintf(fLoader, "%s", buffer);
    fclose(fLoader);
    SQLFreeEnv(henv);
    exit(0);
    return 0;
}
//=====
// Function name: LoadItem
//
//=====
void LoadItem()
{
    int i;
    long i_id;
    long i_im_id;
    char i_name[I_NAME_LEN+1];
    double i_price;
    char i_data[I_DATA_LEN+1];
    char name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcphint[128];
    char err_log_path[256];
    // Seed with unique number
    seed(1);
    printf("Loading item table...\n");
    //if build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxitmc1");
    InitString(i_name, I_NAME_LEN+1);
    InitString(i_data, I_DATA_LEN+1);
    sprintf(name, "%s.%s", aptr->database, "item");
    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "item.err");
    rc = bcp_init(i_hdbc1, name, NULL, err_log_path, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {

```

```

        sprintf(bcp hint, "tablock, ROWS_PER_BATCH = 100000) bcp_control(i_hdbc1, BCPHINTS, (void*) bcp hint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);
    }
    i = 0;
    rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, SQL_VARLEN_DATA, "", 1, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    //time_start = (TimeNow() / MILLI);
    time_start = TimeNow();
    item_rows_loaded = 0;
    for (i_id = 1; i_id <= max_items; i_id++)
    {
        i_im_id =
        RandomNumber(1L, 10000L);
        MakeAlphaStringPadded(14, 24, I_NAME_LEN, i_name);
        i_price = ((float)
        RandomNumber(100L, 10000L))/100.0;
        MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);
        rc =
        bcp_sendrow(i_hdbc1);
        if (rc != SUCCEEDED)
            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()
{
    int            i;
    long           w_id;
    char           w_name[W_NAME_LEN+1];
    char           w_street_1[ADDRESS_LEN+1];
    char           w_street_2[ADDRESS_LEN+1];
    char           w_city[ADDRESS_LEN+1];
    char           w_state[STATE_LEN+1];
    char           w_zip[ZIP_LEN+1];
    double         w_tax;
    double         w_ytd;
    char           name[20];
    long           time_start;
    RETCODE        rc;
    DBINT          rcint;
    char           bcphint[128];
    char           err_log_path[256];

    // Seed with unique number
    seed(2);

    printf("Loading warehouse table...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        BuildIndex("idxwarc");

    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,"whouse.err");
    sprintf(err_log_path,"%swhouse%d.err",
aptr->log_path,aptr->starting_warehouse);
    printf("warehouse error file
=%s\n",err_log_path);
    rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);

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

    /*printf("Setting Tablock for Warehouse
handle\n");
rc = bcp_control(w_hdbc1, BCPHINTS,
"tablock");
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);
}

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *)
&w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
w_name, 0, W_NAME_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
w_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
w_street_2, 0, ADDRESS_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
w_city, 0, ADDRESS_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
w_state, 0, STATE_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_zip,
0, ZIP_LEN, NULL, 0, 0, ++);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    //time_start = (TimeNow() / MILLI);
    time_start = TimeNow();

    warehouse_rows_loaded = 0;

    for (w_id = (long)aptr-
>starting_warehouse; w_id <= (long) aptr-
>num_warehouses; w_id++)
    {
        MakeAlphaStringPadded(6,10,
W_NAME_LEN, w_name);

        MakeAddress(w_street_1,
w_street_2, w_city, w_state, w_zip);

        w_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

        w_ytd = 300000.00;

        rc =
bcp_sendrow(w_hdbc1);

```

```

    if (rc != 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("idxwarc");

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

    District();
    Stock();
}

//=====
//
// Function : District
//
//=====
void District()
{
    int            i;
    short          d_id;
    long           d_w_id;
    char           d_name[D_NAME_LEN+1];
    char           d_street_1[ADDRESS_LEN+1];
    char           d_street_2[ADDRESS_LEN+1];
    char           d_city[ADDRESS_LEN+1];
    char           d_state[STATE_LEN+1];
    char           d_zip[ZIP_LEN+1];
    double         d_tax;
    double         d_ytd;
    char           name[20];
    long           d_next_o_id;
    long           time_start;
    long           w_id;
    RETCODE        rc;
    DBINT          rcint;
    char           bcphint[128];
    char           err_log_path[256];

    // Seed with unique number
    seed(4);

    printf("Loading district table...\n");

    // build index before load
    if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        BuildIndex("idxdisc");

    InitString(d_name, D_NAME_LEN+1);
    InitAddress(d_street_1, d_street_2,
d_city, d_state, d_zip);

```

```

        sprintf(name, "%s..%s", apr->database,
"district");

//      strcpy(err_log_path, apr->log_path);
//      strcat(err_log_path, "district.err");
        sprintf(err_log_path, "%sdistrict%d.err", a
ptr->log_path, apr->starting_warehouse);

        rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        if ((apr->build_index == 1) && (apr-
>index_order == 1))
        {
            sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (apr->num_warehouses
* 10));

            rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcphint);
            if (rc != SUCCEEDED)

                HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&d_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&d_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&d_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
d_name, 0, D_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
d_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
d_street_2, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_city,
0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
d_state, 0, STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip,
0, ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)

```

```

            HandleErrorDBC(w_hdbc1);
        d_ytd = 30000.0;

        d_next_o_id = orders_per_district+1;

        //time_start = (TimeNow() / MILLI);
        time_start = TimeNow();

        for (w_id = (long) apr-
>starting_warehouse; w_id <= (long) apr-
>num_warehouses; w_id++)
        {
            d_w_id = w_id;

            for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
            {

                MakeAlphaStringPadded(6,10,D_NAME_L
EN, 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 ((apr->build_index == 1) && (apr-
>index_order == 0))
                BuildIndex("idxdisc1");

            return;
        }

//=====
//
// Function : Stock
//=====
void Stock()
{
    int          i;
    long         s_i_id;
    long         s_w_id;
    float        PercentDone;
    short        s_quantity;
    char         s_dist_01[S_DIST_LEN+1];

```

```

    char         s_dist_02[S_DIST_LEN+1];
    char         s_dist_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 ((apr->build_index == 1) && (apr-
>index_order == 1))
        BuildIndex("idxstkcd");

    sprintf(name, "%s..%s", apr->database,
"stock");

    //rc = bcp_init(w_hdbc1, name, NULL,
"logs\\stock.err", DB_IN);
    //      strcpy(err_log_path, apr->log_path);
    //      strcat(err_log_path, "stock.err");
    //      sprintf(err_log_path, "%sstock%d.err", ap
tr->log_path, apr->starting_warehouse);

    rc = bcp_init(w_hdbc1, name, NULL,
err_log_path, DB_IN);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    if ((apr->build_index == 1) && (apr-
>index_order == 1))
    {
        sprintf(bcphint, "tablock,
ROWS_PER_BATCH = %u", (apr->num_warehouses
* 100000));
        rc = bcp_control(w_hdbc1,
BCPHINTS, (void*) bcphint);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *)
&s_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);

```

```

        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
&s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_data, 0, SQL_VARLEN_DATA, "", 1, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *)
s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, ++);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt =
0;

        //time_start = (TimeNow() / MILLI);
        time_start = TimeNow();

        printf("...Loading stock table\n");

        for (s_i_id=1; s_i_id <= max_items;
s_i_id++)
        {
            for (s_w_id = (long)aptr-
>starting_warehouse; s_w_id <= (long) aptr-
>num_warehouses; s_w_id++)
            {

```

```

                s_quantity =
(short)RandomNumber(10L,100L); len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                len =
MakeOriginalAlphaString(26,50, S_DATA_LEN,
s_data,10);

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

                stock_rows_loaded++;

                CheckForCommit_Big(w_hdbc1,
w_hstmt1, stock_rows_loaded, "stock", &time_start);

            } //display progress every
10%
            // if (!((float)s_i_id,10) {
                if ((s_i_id % 10) == 0) {
                    PercentDone
= (float) s_i_id *100.0 / (float) max_items;
                    printf("Processed Item %ld, Percent
Done = %.2f\n",s_i_id,PercentDone);
                }
            }

            rcint = bcp_done(w_hdbc1);
            if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

            printf("Finished loading stock table.\n");

            SQLFreeStmt(w_hstmt1, SQL_DROP);
            SQLDisconnect(w_hdbc1);
            SQLFreeConnect(w_hdbc1);

            // if build index after load...
            if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
                BuildIndex("idxstkcl");

            return;
        }
    }
    //=====
    =====
    //

```

```

// Function : LoadCustomer
//=====
//=====
void LoadCustomer()
{
    LOADER_TIME_STRUCT
customer_time_start;
    LOADER_TIME_STRUCT
history_time_start;
    long
        w_id;
    short
        d_id;
        DWORD
dwThreadId[MAX_CUSTOMER_THREADS
];
        HANDLE
hThread[MAX_CUSTOMER_THREADS];
        char
        name[20];
        RETCODE
        rc;
        DBINT
        rcint;
        char
        bcphint[128];
        char
        cmd[256];
        int
        num_procs;
        char
        err_log_path_cust[256];
        char
        err_log_path_hist[256];

        // Seed with unique number
        seed(5);

        printf("Loading customer and history
tables...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr-
>index_order == 1))
        {
            BuildIndex("idxcusc");
            // 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("idxhisc");
        }

        // 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 != SUCCEEDED)
            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 != SUCCEEDED)
                HandleErrorDBC(c_hdbc1);
        }

        sprintf(name, "%s..%s", aptr->database,
"history");
        // rc = bcp_init(c_hdbc2, name, NULL,
"logs\history.err", DB_IN);
        // strcpy(err_log_path_hist, aptr-
>log_path);
        // strcat(err_log_path_hist, "history.err");
        // sprintf(err_log_path_cust, "%shistory%d.
err", aptr->log_path, aptr->starting_warehouse);
        rc = bcp_init(c_hdbc2, name, NULL,
err_log_path_hist, DB_IN);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        sprintf(bcphint, "tablock");
        rc = bcp_control(c_hdbc2, BCPHINTS,
(void*) bcphint);
        if (rc != SUCCEEDED)
            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 =
%d, 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
                {
                    printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
                }

                if
                {
                    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("idxcusc1");
            // check the number of
processors on this system
            // if 8 or more processors,
then build index on History.
            // if less than 8 processors,
do not build the index
            num_procs = atoi(getenv(
"NUMBER_OF_PROCESSORS" ));
            if (num_procs >= 8)
                BuildIndex("idxhiscl");
            }

            // build non-clustered index
            if (aptr->build_index == 1)
                BuildIndex("idxcuscnc");

            // Output the NURAND used for the
loader into C_FIRST for C_ID = 1,
            // C_W_ID = 1, and C_D_ID = 1
            /* sprintf(cmd, "osql -S%s -U%s -P%s -
d%s -e -Q"update customer set c_first = 'C_LOAD =
%d' where c_id = 1 and c_w_id = 1 and c_d_id = 1"
> %snurand_load.log",

                aptr->server,

                aptr->user,

                aptr->password,

                aptr->database,

                LOADER_NURAND_C,

                aptr->log_path);

            system(cmd);

        */

        SQLFreeStmt(c_hstmt1, SQL_DROP);
        SQLDisconnect(c_hdbc1);
        SQLFreeConnect(c_hdbc1);

        SQLFreeStmt(c_hstmt2, SQL_DROP);
        SQLDisconnect(c_hdbc2);
        SQLFreeConnect(c_hdbc2);

        return;
    }
}

```

```

//=====
//=====
//
// Function : CustomerBufInit
//
//=====
//=====
void CustomerBufInit()
{
    long i;

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_id = 0;
        customer_buf[i].c_d_id =
0;
        customer_buf[i].c_w_id =
0;

        strcpy(customer_buf[i].c_first,"");
        strcpy(customer_buf[i].c_middle,"");
        strcpy(customer_buf[i].c_last,"");
        strcpy(customer_buf[i].c_street_1,"");
        strcpy(customer_buf[i].c_street_2,"");
        strcpy(customer_buf[i].c_city,"");
        strcpy(customer_buf[i].c_state,"");
        strcpy(customer_buf[i].c_zip,"");
        strcpy(customer_buf[i].c_phone,"");
        strcpy(customer_buf[i].c_credit,"");

        customer_buf[i].c_credit_lim = 0;
        customer_buf[i].c_discount
= (float) 0;

        strcpy(customer_buf[i].c_balance,"");

        customer_buf[i].c_ytd_payment = 0;
        customer_buf[i].c_payment_cnt = 0;
        customer_buf[i].c_delivery_cnt = 0;

        strcpy(customer_buf[i].c_data,"");

        customer_buf[i].h_amount
= 0;

        strcpy(customer_buf[i].h_data,"");
    }
}
//=====
//=====
//
// Function : CustomerBufLoad

```

```

// Fills shared buffer for HISTORY and CUSTOMER
//=====
//=====
void CustomerBufLoad(int d_id, long w_id)
{
    long
i;
    CUSTOMER_SORT_STRUCT
c[CUSTOMERS_PER_DISTRICT];

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i,
c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_N
URAND_C), c[i].c_last);

        MakeAlphaStringPadded(8,16,FIRST_NA
ME_LEN, c[i].c_first);

        c[i].c_id = i+1;
    }

    printf("...Loading customer buffer for:
d_id = %ld, w_id = %ld\n",
d_id, w_id);

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_d_id =
d_id;
        customer_buf[i].c_w_id =
w_id;
        customer_buf[i].h_amount
= 10.0;
        customer_buf[i].c_ytd_payment = 10.0;
        customer_buf[i].c_payment_cnt = 1;
        customer_buf[i].c_delivery_cnt = 0;
        customer_buf[i].c_id =
c[i].c_id;
        strcpy(customer_buf[i].c_first,
c[i].c_first);
        strcpy(customer_buf[i].c_last,
c[i].c_last);
        customer_buf[i].c_middle[0] = 'O';
        customer_buf[i].c_middle[1] = 'E';

        MakeAddress(customer_buf[i].c_street_1
,
customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,
customer_buf[i].c_zip);
        MakeNumberString(16, 16,
PHONE_LEN, customer_buf[i].c_phone);

        if (RandomNumber(1L,
100L) > 10)

```

```

        customer_buf[i].c_credit[0] = 'G';
        customer_buf[i].c_credit[0] = 'B';
        customer_buf[i].c_credit[1]
= 'C';
        customer_buf[i].c_credit_lim = 50000.0;
        customer_buf[i].c_discount
= ((float) RandomNumber(0L, 5000L)) / 10000.0;

        strcpy(customer_buf[i].c_balance,"-
10.0");
        MakeAlphaStringPadded(300, 500,
C_DATA_LEN, customer_buf[i].c_data);

        // Generate HISTORY data

        MakeAlphaStringPadded(12, 24,
H_DATA_LEN, customer_buf[i].h_data);
    }
}
//=====
//=====
//
// Function : LoadCustomerTable
//
//=====
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT
*customer_time_start)
{
    long
i;
    long
c_id;
    short
c_d_id;
    long
c_w_id;
    char
c_first[FIRST_NAME_LEN+1];
    char
c_middle[MIDDLE_NAME_LEN+1];
    char
c_last[LAST_NAME_LEN+1];
    char
c_street_1[ADDRESS_LEN+1];
    char
c_street_2[ADDRESS_LEN+1];
    char
c_city[ADDRESS_LEN+1];
    char
c_state[STATE_LEN+1];
    char
c_zip[ZIP_LEN+1];
    char
c_phone[PHONE_LEN+1];
    char
c_credit[CREDIT_LEN+1];
    double
c_credit_lim;
    double
c_discount;
    char
c_balance[6];
    double
c_ytd_payment;
    short
c_payment_cnt;
    short
c_delivery_cnt;
    char
c_data[C_DATA_LEN+1];
    char
c_since[C_SINCE_LEN+1];
    RETCODE
rc;

    i = 0;
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);

```

```

        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0,
LAST_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_first,
0, FIRST_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0,
CREDIT_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5,
NULL, 0, SQLCHARACTER, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment,
0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *)
&c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0,
PHONE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *)
&c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *)
c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0,
C_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)

```

```

            HandleErrorDBC(c_hdbc1);
        for (i = 0; i < customers_per_district; i++)
        {
            customer_buff[i].c_id =
            customer_buff[i].c_d_id =
            customer_buff[i].c_w_id =
            strcpy(c_first,
customer_buff[i].c_first);
            strcpy(c_middle,
customer_buff[i].c_middle);
            strcpy(c_last,
customer_buff[i].c_last);
            strcpy(c_street_1,
customer_buff[i].c_street_1);
            strcpy(c_street_2,
customer_buff[i].c_street_2);
            strcpy(c_city,
customer_buff[i].c_city);
            strcpy(c_state,
customer_buff[i].c_state);
            strcpy(c_zip,
customer_buff[i].c_zip);
            strcpy(c_phone,
customer_buff[i].c_phone);
            strcpy(c_credit,
customer_buff[i].c_credit);
            FormatDate(&c_since);
            c_credit_lim =
customer_buff[i].c_credit_lim;
            c_discount =
customer_buff[i].c_discount;
            strcpy(c_balance,
customer_buff[i].c_balance);
            c_ytd_payment =
customer_buff[i].c_ytd_payment;
            c_payment_cnt =
customer_buff[i].c_payment_cnt;
            c_delivery_cnt =
customer_buff[i].c_delivery_cnt;
            strcpy(c_data,
customer_buff[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;

        i = 0;
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *)
&c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *)
&h_date, 0, H_DATE_LEN, NULL, 0, SQLCHARACTER,
++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0,
H_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(c_hdbc2);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id =
customer_buff[i].c_id;
            c_d_id =
customer_buff[i].c_d_id;
            c_w_id =
customer_buff[i].c_w_id;
            h_amount =
customer_buff[i].h_amount;
            strcpy(h_data,
customer_buff[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
short
w_id;
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.e
rr",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");

        // rc = bcp_init(o_hdbc2, name, NULL,
"logs\\neword.err", DB_IN);
        // strcpy(err_log_path_nord,aptr-
>log_path);
        // strcat(err_log_path_nord,"neword.err");
        sprintf(err_log_path_nord,"%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");

            // rc = bcp_init(o_hdbc3,
name, NULL, "logs\\ordline.err", DB_IN);
            // strcpy(err_log_path_ordl,aptr-
>log_path);
            // strcat(err_log_path_ordl,"ordline.err");
            sprintf(err_log_path_ordl,"%sordline%d.
err",aptr->log_path,aptr->starting_warehouse);
            rc = bcp_init(o_hdbc3, name, NULL,
err_log_path_ordl, DB_IN);
            if (rc != 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 here...
                // start Orders
table thread

                printf("...Loading Order Table for: d_id =
%d, w_id = %d\n", d_id, w_id);

                hThread[0] =
CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE)
LoadOrdersTable,
&orders_time_start,
0,
&dwThreadId[0]);

                if (hThread[0]
== NULL)
                {
                    printf("Error, failed in creating creating
thread = 0.\n");
                    exit(-1);
                }

                // start
NewOrder table thread

                printf("...Loading New-Order Table for:
d_id = %d, w_id = %d\n", d_id, w_id);

                hThread[1] =
CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE)
LoadNewOrderTable,
&new_order_time_start,
0,

```

```

        &dwThreadID[1]);
    if (hThread[1]
== NULL)
    {
        printf("Error, failed in creating creating
thread = 1.\n");
        exit(-1);
    }
    // start Order-
Line table thread
    printf("...Loading Order-Line Table for:
d_id = %ld, w_id = %ld\n", d_id, w_id);
    hThread[2] =
CreateThread(NULL,
    0,
    (LPTHREAD_START_ROUTINE)
LoadOrderLineTable,
    &order_line_time_start,
    0,
    &dwThreadID[2]);
    if (hThread[2]
== NULL)
    {
        printf("Error, failed in creating creating
thread = 2. LastError = %d\n", GetLastError());
        exit(-1);
    }
    WaitForSingleObject( hThread[0],
INFINITE );
    WaitForSingleObject( hThread[1],
INFINITE );
    WaitForSingleObject( hThread[2],
INFINITE );
    if
(CloseHandle(hThread[0]) == FALSE)
    {
        printf("Error, failed in closing Orders
thread handle with errno: %d\n", GetLastError());
    }
    if
(CloseHandle(hThread[1]) == FALSE)
    {
        printf("Error, failed in closing NewOrder
thread handle with errno: %d\n", GetLastError());
    }

```

```

    if
(CloseHandle(hThread[2]) == FALSE) {
        printf("Error, failed in closing OrderLine
thread handle with errno: %d\n", GetLastError());
    }
    }
    printf("Finished loading orders.\n");
    return;
}
//=====
//
// Function : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and
ORDERLINE
//
//=====
void OrdersBufInit()
{
    int i;
    int j;
    for (i=0;i<orders_per_district;i++)
    {
        orders_buf[i].o_id = 0;
        orders_buf[i].o_d_id = 0;
        orders_buf[i].o_w_id = 0;
        orders_buf[i].o_c_id = 0;
        orders_buf[i].o_carrier_id =
0;
        orders_buf[i].o_ol_cnt = 0;
        orders_buf[i].o_all_local =
0;
        for (j=0;j<=14;j++)
        {
            orders_buf[i].o_ol[j].ol = 0;
            orders_buf[i].o_ol[j].ol_i_id = 0;
            orders_buf[i].o_ol[j].ol_supply_w_id = 0;
            orders_buf[i].o_ol[j].ol_quantity = 0;
            orders_buf[i].o_ol[j].ol_amount = 0;
            strcpy(orders_buf[i].o_ol[j].ol_dist_info,"
");
        }
    }
//=====
//
// Function : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and
ORDERLINE
//
//=====
void OrdersBufLoad(short d_id, long w_id)

```

```

{
    int cust[ORDERS_PER_DISTRICT+1];
    long o_id;
    long ol;
    printf("...Loading Order Buffer for: d_id
= %ld, w_id = %ld\n",
        d_id, w_id);
    GetPermutation(cust,
orders_per_district);
    for
(o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and
NEW-ORDER data
        orders_buf[o_id].o_d_id =
d_id;
        orders_buf[o_id].o_w_id =
w_id;
        orders_buf[o_id].o_id =
o_id+1;
        orders_buf[o_id].o_c_id =
cust[o_id+1];
        orders_buf[o_id].o_ol_cnt
= (short)RandomNumber(5L, 15L);
        if (o_id < first_new_order)
        {
            orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
            orders_buf[o_id].o_all_local = 1;
        }
        else
        {
            orders_buf[o_id].o_carrier_id = 0;
            orders_buf[o_id].o_all_local = 1;
        }
        for (ol=0;
ol<orders_buf[o_id].o_ol_cnt; ol++)
        {
            orders_buf[o_id].o_ol[ol].ol = ol+1;
            orders_buf[o_id].o_ol[ol].ol_i_id =
RandomNumber(1L, max_items);
            orders_buf[o_id].o_ol[ol].ol_supply_w_id
= w_id;
            orders_buf[o_id].o_ol[ol].ol_quantity =
5;
            MakeAlphaString(24, 24,
OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);
            // Generate
ORDER-LINE data
            if (o_id <
first_new_order)
            {
                orders_buf[o_id].o_ol[ol].ol_amount = 0;
            }
            // Added to insure ol_delivery_d set
properly during load

```



```

FormatDate(&orders_buf[o_id].o_o[ol].o_o_l_delivery_d);
}
else
{
orders_buf[o_id].o_o[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_o[ol].ol_delive
ry_d,"1899-12-31 00:00:00.000");
}
}
}

//=====
//
// Function : LoadOrdersTable
//
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT
*orders_time_start)
{
int i;
long o_id;
short o_d_id;
long o_w_id;
long o_c_id;
short o_carrier_id;
short o_ol_cnt;
short o_all_local;
char o_entry_d[O_ENTRY_D_LEN+1];
RETCODE rc;
DBINT rcint;

// bind ORDER data
i = 0;
rc = bcp_bind(o_hdbc1, (BYTE *) &o_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
}
}

```

```

if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);
rc = bcp_bind(o_hdbc1, (BYTE *)
&o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0,
SQLCHARACTER, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc1);

for (i = 0; i < orders_per_district; i++)
{
orders_buf[i].o_id =
orders_buf[i].o_d_id =
orders_buf[i].o_w_id =
orders_buf[i].o_c_id =
orders_buf[i].o_carrier_id =
orders_buf[i].o_ol_cnt =
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);
}

if ((o_w_id == apr->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");
}
}
//=====
//
//=====
rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
}
}

```

```

// Function : LoadNewOrderTable
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT
*new_order_time_start)
{
long i;
long o_id;
short o_d_id;
long o_w_id;
RETCODE rc;
DBINT rcint;

// Bind NEW-ORDER data
i = 0;
rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc2);
rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc2);
rc = bcp_bind(o_hdbc2, (BYTE *)
&o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
if (rc != SUCCEEDED)
HandleErrorDBC(o_hdbc2);

for (i = first_new_order; i <
last_new_order; i++)
{
orders_buf[i].o_id =
orders_buf[i].o_d_id =
orders_buf[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);
if ((o_w_id == apr->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          long          i;
    long          long          j;
    long          o_id;
    short         o_d_id;
    long          o_w_id;
    double        ol;
    long          ol_i_id;
    long          ol_supply_w_id;
    short         ol_quantity;
    double        ol_amount;
    char          ol_dist_info[DIST_INFO_LEN+1];
    char          ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE       rc;
    DBINT         rcint;

    // bind ORDER-LINE data
    i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0,
SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *)
&ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 0, ++i);

```

```

    if (rc != SUCCEEDED) HandleErrorDBC(o_hdbc3);
    for (i = 0; i < orders_per_district; i++)
    {
        orders_buf[i].o_id =
        orders_buf[i].o_d_id;
        orders_buf[i].o_w_id;
        orders_buf[i].o_ol_cnt; j++)
        {
            orders_buf[i].o_ol[j].ol =
            orders_buf[i].o_ol[j].ol_i_id =
            orders_buf[i].o_ol[j].ol_supply_w_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].o
.ol_delivery_d);

            strcpy(ol_dist_info,orders_buf[i].o_ol[j].o
L_dist_info);

            rc =
            bcp_sendrow(o_hdbc3);
            if (rc !=
            SUCCEEDED)
                HandleErrorDBC(o_hdbc3);
            order_line_rows_loaded++;

            CheckForCommit_Big(o_hdbc3,
o_hstmt3, order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
        }
        if ((o_w_id == apr->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 ((apr->build_index ==
1) && (apr->index_order == 0))
                BuildIndex("idxodcl");
        }
    }

```

```

//=====
// 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 % apr->batch) )
    {
        time_end = (TimeNow() /
MILLI);
        time_end = TimeNow();
        time_diff = time_end -
*time_start;

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

        *time_start = time_end;
        bcp_batch(hdbc);
    }
}

return;
}

//=====

```

```

//
// Function : CheckForCommit_Big
//
//=====
//=====
void CheckForCommit_Big(HDBC hdbc,
                        HSTMT hstmt,
                        double rows_loaded,
                        char *table_name,
                        long *time_start)
{
    long    time_end, time_diff;
    if ( !(fmod(rows_loaded,aptr->batch) ) )
    {
        //          time_end = (TimeNow() /
        MILLI);
        time_end = TimeNow();
        time_diff = time_end -
*time_start;
        printf("-> Loaded %ld rows
into %s in %ld ms - Total = %.0f (%.2f rpms)\n",
                aptr->batch,
                table_name,
                time_diff,
                rows_loaded,
                (float) aptr->batch / (time_diff ?
time_diff : 1L));
        *time_start = time_end;
        bcp_batch(hdbc);
    }
    return;
}
//=====
//=====
// Function : OpenConnections
//
//=====
//=====
void OpenConnections()
{
    RETCODE    rc;
    char
    szDriverString[300];
    char
    szDriverStringOut[1024];
    SQLSMALLINT
    cbDriverStringOut;
    SQLAllocHandle(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv );
    SQLSetEnvAttr(henv,
SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3,
0 );

```

```

    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &i_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &w_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &c_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &c_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc1);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc2);
    SQLAllocHandle(SQL_HANDLE_DBC,
henv , &o_hdbc3);
    SQLSetConnectAttr(i_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(w_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(c_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc1,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc2,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc3,
SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    // Open connections to SQL Server
    // Connection 1
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,
                aptr->server,
                aptr->user,
                aptr->password,
                aptr->database );
    printf("opening db connections, open
string = %s\n",szDriverString);
    rc = SQLSetConnectOption (i_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = SQLDriverConnect ( i_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT
);

```

```

    if ( (rc != SUCCEEDED) &&
SQL_SUCCESS_WITH_INFO ) )
    {
        HandleErrorDBC(i_hdbc1);
        printf("TPC-C Loader
aborted!\n");
        exit(9);
    }
    // Connection 2
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,
                aptr->server,
                aptr->user,
                aptr->password,
                aptr->database );
    rc = SQLSetConnectOption (w_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = SQLDriverConnect ( w_hdbc1,
        NULL,
        (SQLCHAR*)&szDriverString[0] ,
        SQL_NTS,
        (SQLCHAR*)&szDriverStringOut[0],
        sizeof(szDriverStringOut),
        &cbDriverStringOut,
        SQL_DRIVER_NOPROMPT );
    if ( (rc != SUCCEEDED) &&
(rc !=
SQL_SUCCESS_WITH_INFO) )
    {
        HandleErrorDBC(w_hdbc1);
        printf("TPC-C Loader
aborted!\n");
        exit(9);
    }
    // Connection 3
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,
                aptr->server,
                aptr->user,
                aptr->password,
                aptr->database );
    rc = SQLSetConnectOption (c_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEEDED)

```

```

        HandleErrorDBC(c_hdbc1);
rc = SQLDriverConnect ( c_hdbc1,
                        NULL,
(SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
(SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc !=
SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 4
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database );

rc = SQLSetConnectOption (c_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

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) )
{
    HandleErrorDBC(c_hdbc2);

```

```

        printf("TPC-C Loader
aborted!\n");
        exit(9);
    }

    // Connection 5
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database );

rc = SQLSetConnectOption (o_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

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) )
{
    HandleErrorDBC(o_hdbc1);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 6
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database );

rc = SQLSetConnectOption (o_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc2);

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) )
{
    HandleErrorDBC(o_hdbc2);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}

// Connection 7
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=
%s" ,

        aptr->server,
        aptr->user,
        aptr->password,
        aptr->database );

rc = SQLSetConnectOption (o_hdbc3,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc3);

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) )
{
    HandleErrorDBC(o_hdbc3);
    printf("TPC-C Loader
aborted!\n");
    exit(9);
}
}

```

```

//=====
//
// Function name: BuildIndex
//
//=====
void BuildIndex(char *index_script)
{
    char cmd[256];

    printf("Starting index creation:
%s\n",index_script);

    sprintf(cmd, "osql -S%s -U%s -P%s -e -
i%s\\%s.sql > %s%s.log",

aptr->server,
aptr->user,
aptr->password,
aptr->index_script_path,
index_script,
aptr->log_path,
index_script);

system(cmd);

printf("Finished index creation:
%s\n",index_script);
}

//=====
//
// Function name: HandleErrorDBC
//
//=====
void HandleErrorDBC (SQLHDBC hdbc1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;
SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
char timebuf[128];
char datebuf[128];
char err_log_path[256];
FILE *fp1;

i = 1;
while ((rc2 =
SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState
, &NativeError,
Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
{
    sprintf( szLastError , "%s" ,
Msg );

    _strtime(timebuf);
    _strdate(datebuf);
}
}

```

```

printf( "[%s : %s]
%s\n=>SQLState: %s\n" , datebuf, timebuf,
szLastError, SqlState);
// strcpy(err_log_path,aptr-
>log_path);
// strcat(err_log_path,"tpccldr.err");

sprintf(err_log_path,"%stpcldr%d.err",aptr-
>log_path,aptr->starting_warehouse);
fp1 =
fopen(err_log_path,"a+");
if (fp1 == NULL)

printf("ERROR: Unable to open errorlog
file.\n");
else
{
    fprintf(fp1,
"%s : %s] %s\nSQLState: %s\n" , datebuf, timebuf,
szLastError, SqlState);
    fclose(fp1);
}
i++;
}

//=====
//
// Function : HandleErrorSTMT
//
//=====
void HandleErrorSTMT (HSTMT hstmt1)
{
    SQLCHAR SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
SQLINTEGER NativeError;
SQLSMALLINT i, MsgLen;
SQLRETURN rc2;
char timebuf[128];
char datebuf[128];
char err_log_path[256];
FILE *fp1;

i = 1;
while ((rc2 =
SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i,
SqlState , &NativeError,
Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
{
    if (total_db_errors >=
MAX_SQL_ERRORS)
    {
        printf(">>>> Maximum SQL errors of
%d exceeded. Terminating
TPCCCLR.<<<<<\n",total_db_errors);
        exit(9);
    }
    total_db_errors++;

    sprintf( szLastError , "%s" ,
Msg );

    _strtime(timebuf);
    _strdate(datebuf);
}
}

```

```

printf( "[%s : %s]
%s\nSQLState: %s\n" , datebuf, timebuf, szLastError,
SqlState);
// strcpy(err_log_path,aptr-
>log_path);
// strcat(err_log_path,"tpccldr.err");

sprintf(err_log_path,"%stpcldr%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;
}

time.c

// File: TIME.C
//
// Microsoft TPC-C Kit Ver. 4.42
//
// Copyright Microsoft, 1996, 1997, 1998,
1999, 2000, 2001, 2002, 2003
// Purpose: Source file for time
functions

// Includes
#include "tpcc.h"

```

```
// Globals
static long start_sec;

//=====
//=====
//
// Function name: TimeNow
//
//=====
//=====

long TimeNow()
{
    long          time_now;
    struct        _timeb el_time;

#ifdef DEBUG
    printf("[%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
//
//      Microsoft TPC-C Kit Ver. 4.51
//
//      Copyright Microsoft, 1996, 1997, 1998,
//      1999, 2000, 2001, 2002, 2003
//      Purpose:      Source file for database
//      loader string functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//=====
//
// Function name: MakeAddress
//
//=====
//=====

void MakeAddress(char *street_1,
char *street_2,
char *city,
char *state,
char *zip)
{
```

```
#ifdef DEBUG
    printf("[%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[] =
    {
        "PRI", "PRES", "BAR", "OUGHT", "ABLE",
        "ATION", "EING", "ESE", "ANTI", "CALLY",
    };

#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 <%ld> 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[] =
    "0123456789ABCDEFGHIJKLMNQRSTUUVWXYZabcde
fghijklmnopqrstuvwxyz";
    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++)
        str[i] =
chArray[RandomNumber(0, chArrayMax)];
    str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int minLen, int maxLen,
int padLen, char *str)
{
    int          len;
    int          i;
```

```

char cc = 'a';
static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijkmnopqrstuvwxyz";
static int chArrayMax =
61;

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

len = RandomNumber(minLen, maxLen);

for (i=0; i<len; i++)
str[i] =
chArray[RandomNumber(0,chArrayMax)];
if (len < padLen)
memset(str+len, ' ', padLen
- len);
str[padLen] = 0;
return padLen;
}

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

#ifdef DEBUG
printf("[%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 < 8)
{
printf("MakeOriginalAlphaString: string
length must be >= 8\n");
exit(-1);
}
}

```

```

int MakeAlphaString(x,y, z, str);

val = RandomNumber(1,100);
if (val <= percent)
{
start = RandomNumber(0,
len - 8);
strncpy(str + start,
"ORIGINAL", 8);
}

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

return len;
}

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

//MakeNumberString is always called
MakeZipNumberString(16, 16, 16, string)

memset(str, '0', 16);
itoa(RandomNumber(0, 99999999), tmp,
10);
memcpy(str, tmp, strlen(tmp));

itoa(RandomNumber(0, 99999999), tmp,
10);
memcpy(str+8, tmp, strlen(tmp));

str[16] = 0;

return 16;
}

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

//MakeZipNumberString is always called
MakeZipNumberString(9, 9, 9, string)

strcpy(str, "000011111");

itoa(RandomNumber(0, 9999), tmp, 10);
memcpy(str, tmp, strlen(tmp));
}

```

```

} return 9;

//=====
//
// Function name: InitString
//
//=====
void InitString(char *str, int len)
{
#ifdef DEBUG
printf("[%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("[%d]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("[%d]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 %d &
%d ==> %d\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. 4.51
// 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,
TPCCCLDR_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
= DEF_LDPACKSIZE;
    pargs->starting_warehouse
= DEF_STARTING_WAREHOUSE;
    pargs->build_index
= BUILD_INDEX;
    pargs->index_order
= 0;
    pargs->index_script_path =
INDEX_SCRIPT_PATH;
    pargs->scale_down
= SCALE_DOWN;

    /* 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();
    }

    ptr = argv[i];

    switch (ptr[1])
    {
        case '?': /* Fall through

*/

        GetArgsLoaderUsage();

        break;

        case 'D':

        pargs->database = ptr+2;

        break;

        case 'P':

        pargs->password = ptr+2;

        break;

        case 'S':

        pargs->server = ptr+2;

        break;

        case 'U':

        pargs->user = ptr+2;

        break;

        case 'b':

        pargs->batch = atol(ptr+2);

        break;

        case 'W':

        pargs->num_warehouses = atol(ptr+2);

        break;

        case 's':

        pargs->starting_warehouse =
atol(ptr+2);

        break;

        case 't':

        {

        pargs->tables_all = FALSE;

        if (strcmp(ptr+2,"item") == 0)

        pargs->table_item = TRUE;

        else if (strcmp(ptr+2,"warehouse") ==
0)

        pargs->table_warehouse =
TRUE;

        else if (strcmp(ptr+2,"customer") == 0)

```

```

TRUE;
    pargs->table_customer =
else if (strcmp(ptr+2,"orders") == 0)
    pargs->table_orders =
TRUE;
else
{
printf("\nUnrecognized command");
GetArgsLoaderUsage();
exit(1);
}

break;
}

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

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

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

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

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

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

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

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
-----\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) DEFLDPACKSIZE);
printf("-L Loader BCP Log Path
%s\n", LOADER_LOG_PATH);
printf("-f 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("-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,
DICTIONARY, and STORED PROCEDURES,
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. 4.68
--
-- Copyright Microsoft, 2006
--
-- Sets database options for load
--
-----
USE master
GO

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
GO

EXEC sp_dboption tpcc,'trunc. log on chkpt.',TRUE
GO

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION
OFF
GO

ALTER DATABASE tpcc SET PAGE_VERIFY NONE
GO

USE tpcc
GO

CHECKPOINT
GO

```

DBOPT2 . SQL

```

-----
--
-- File: DBOPT2.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
--
-- Copyright Microsoft, 2006
--

```

```

-- Sets database options after load
--
--
-----
ALTER DATABASE tpcc SET RECOVERY FULL
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE @msg varchar(50)

-----
-- OPTIONS FOR SQL SERVER 2000 --
-- Set option values for user-defined indexes --
-----

SET @msg = ''
PRINT @msg
SET @msg = 'Setting SQL Server indexoptions'
PRINT @msg
SET @msg = ''
PRINT @msg

EXEC sp_indexoption 'customer',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'district',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'warehouse',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'stock',
'DisAllowPageLocks', TRUE
EXEC sp_indexoption 'order_line',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'orders',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'new_order',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowRowLocks', TRUE
EXEC sp_indexoption 'item',
'DisAllowPageLocks', FALSE
GO

Print ''
Print '*****'
Print 'Pre-specified Locking Hierarchy:'
Print ' Lockflag = 0 ==> No pre-specified
hierarchy'
Print ' Lockflag = 1 ==> Lock at Page-level then
Table-level'
Print ' Lockflag = 2 ==> Lock at Row-level then
Table-level'
Print ' Lockflag = 3 ==> Lock at Table-level'
Print ''

SELECT name,
lockflags
FROM sysindexes
WHERE object_id('warehouse') = id OR
object_id('district') = id OR
object_id('customer') = id OR
object_id('stock') = id OR
object_id('orders') = id OR
object_id('order_line') = id OR
object_id('history') = id OR

```

```

object_id('item')=id id OR
ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc, 'auto update statistics',
FALSE
EXEC sp_dboption tpcc, 'auto create statistics',
FALSE
GO

DECLARE @db_id int,
@tbl_id int

SET @db_id = DB_ID('tpcc')
SET @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO

```

B.2 Table definitions

Createdatabase.sql

```

--
-- HPTPCC Setup Kit
-- Create Database
--
-- Create main database files
--
-- Files are loaded into the load filegroup
-- Then Create[table]Indexes.sql moves them to the
final filegroup

if exists (select * from sysdatabases where
name='tpcc') drop database tpcc

CREATE DATABASE tpcc
ON PRIMARY
(
NAME = tpccRoot,
FILENAME = "c:\tpcc100kwh.mdf",
SIZE = 12MB,
FILEGROWTH = 0),

FILEGROUP cs_fg
(NAME=cs1, FILENAME="c:\mnt\cs\1",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs2, FILENAME="c:\mnt\cs\2",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs3, FILENAME="c:\mnt\cs\3",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs4, FILENAME="c:\mnt\cs\4",
SIZE=512000MB,FILEGROWTH=0),

```

```

(NAME=cs5, FILENAME="c:\mnt\cs\5",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs6, FILENAME="c:\mnt\cs\6",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs7, FILENAME="c:\mnt\cs\7",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs8, FILENAME="c:\mnt\cs\8",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs9, FILENAME="c:\mnt\cs\9",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs10, FILENAME="c:\mnt\cs\10",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs11, FILENAME="c:\mnt\cs\11",
SIZE=512000MB,FILEGROWTH=0),
(NAME=cs12, FILENAME="c:\mnt\cs\12",
SIZE=512000MB,FILEGROWTH=0),

FILEGROUP misc_fg

(NAME=misc1, FILENAME="c:\mnt\misc\1",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc2, FILENAME="c:\mnt\misc\2",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc3, FILENAME="c:\mnt\misc\3",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc4, FILENAME="c:\mnt\misc\4",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc5, FILENAME="c:\mnt\misc\5",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc6, FILENAME="c:\mnt\misc\6",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc7, FILENAME="c:\mnt\misc\7",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc8, FILENAME="c:\mnt\misc\8",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc9, FILENAME="c:\mnt\misc\9",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc10, FILENAME="c:\mnt\misc\10",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc11, FILENAME="c:\mnt\misc\11",
SIZE=281600MB,FILEGROWTH=0),
(NAME=misc12, FILENAME="c:\mnt\misc\12",
SIZE=281600MB,FILEGROWTH=0),

```

```

FILEGROUP load_fg
(NAME=load1,
FILENAME="c:\mnt\backup\1\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load2,
FILENAME="c:\mnt\backup\2\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load3,
FILENAME="c:\mnt\backup\3\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load4,
FILENAME="c:\mnt\backup\4\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load5,
FILENAME="c:\mnt\backup\5\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load6,
FILENAME="c:\mnt\backup\6\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load7,
FILENAME="c:\mnt\backup\7\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load8,
FILENAME="c:\mnt\backup\8\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load9,
FILENAME="c:\mnt\backup\9\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),

```

```

(NAME=load10,
FILENAME="c:\mnt\backup\10\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load11,
FILENAME="c:\mnt\backup\11\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load12,
FILENAME="c:\mnt\backup\12\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load13,
FILENAME="c:\mnt\backup\13\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load14,
FILENAME="c:\mnt\backup\14\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load15,
FILENAME="c:\mnt\backup\15\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load16,
FILENAME="c:\mnt\backup\16\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load17,
FILENAME="c:\mnt\backup\5tb_1\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0),
(NAME=load18,
FILENAME="c:\mnt\backup\5tb_2\loadfg.ndf",
SIZE=192000MB,FILEGROWTH=0)

```

```

LOG ON
(
NAME = tpcc_log,
FILENAME = "L:",
SIZE = 400000MB,
FILEGROWTH = 0)

```

```

COLLATE Latin1_General_BIN
GO

```

CreateTables.sql

```

-----
-- File: TABLES.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.62
-- Copyright Microsoft, 2005 --
-- Creates TPC-C tables --
-----

```

```

SET ANSI_NULL_DFLT_OFF ON
GO

```

```

USE tpcc
GO

```

```

-----
-- Remove all existing TPC-C tables
-----
if exists ( select name from sysobjects where name =
'warehouse' )
drop table warehouse
go
if exists ( select name from sysobjects where name =
'district' )
drop table district
go

```

```

if exists ( select name from sysobjects where name =
'customer' ) drop table customer
go
if exists ( select name from sysobjects where name =
'history' )
drop table history
go
if exists ( select name from sysobjects where name =
'new_order' )
drop table new_order
go
if exists ( select name from sysobjects where name =
'orders' )
drop table orders
go
if exists ( select name from sysobjects where name =
'order_line' )
drop table order_line
go
if exists ( select name from sysobjects where name =
'item' )
drop table item
go
if exists ( select name from sysobjects where name =
'stock' )
drop table stock
go

```

```

-----
-- Create new tables
-----

```

```

create table warehouse
(
w_id int,
w_ytd money,
w_tax smallmoney,
w_name char(10),
w_street_1 char(20),
w_street_2 char(20),
w_city char(20),
w_state char(2),
w_zip char(9)
) on load_fg
go

```

```

create table district
(
d_id tinyint,
d_w_id int,
d_ytd money,
d_next_o_id int,
d_tax smallmoney,
d_name char(10),
d_street_1 char(20),
d_street_2 char(20),
d_city char(20),
d_state char(2),
d_zip char(9)
) on load_fg
go

```

```

create table customer
(
c_id int,
c_d_id tinyint,
c_w_id int,
c_discount smallmoney,
c_credit_lim money,
c_last char(16),
c_first char(16),
c_credit char(2),
c_balance money,
c_ytd_payment money,
c_payment_cnt smallint,

```

```

c_street_1 char(20),
c_street_2 char(20),
c_city char(20),
c_state char(2),
c_zip char(9),
c_phone char(16),
c_since datetime,
c_middle char(2),
c_data char(500)
) on load_fg
go

```

```

-- Use the following table option if using c_data
varchar(max)
-- sp_tableoption 'customer','large value types out of
row','1'
-- go

```

```

create table history
(
h_c_id int,
h_c_d_id tinyint,
h_c_w_id int,
h_d_id tinyint,
h_w_id int,
h_date datetime,
h_amount smallmoney,
h_data char(24)
) on load_fg
go

```

```

create table new_order
(
no_o_id int,
no_d_id tinyint,
no_w_id int
) on load_fg
go

```

```

create table orders
(
o_id int,
o_d_id tinyint,
o_w_id int,
o_c_id int,
o_carrier_id tinyint,
o_ol_cnt tinyint,
o_all_local tinyint,
o_entry_d datetime
) on load_fg
go

```

```

create table order_line
(
ol_o_id int,
ol_d_id tinyint,
ol_w_id int,
ol_number tinyint,
ol_i_id int,
ol_delivery_d datetime,
ol_amount smallmoney,
ol_supply_w_id int,
ol_quantity smallint,
ol_dist_info char(24)
) on load_fg
go

```

```

create table item
(
i_id int,
i_name char(24),
i_price smallmoney,
i_data char(50),
i_im_id int
) on load_fg

```

```

go

create table stock
(
    s_i_id          int,
    s_w_id          int,
    s_quantity     smallint,
    s_ytd          int,
    s_order_cnt    smallint,
    s_remote_cnt   smallint,
    s_data         char(50),
    s_dist_01      char(24),
    s_dist_02      char(24),
    s_dist_03      char(24),
    s_dist_04      char(24),
    s_dist_05      char(24),
    s_dist_06      char(24),
    s_dist_07      char(24),
    s_dist_08      char(24),
    s_dist_09      char(24),
    s_dist_10      char(24)
) on load_fg
go

```

Clustered/NonClustered Index Creation Scripts

```

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

if exists (select name from sysindexes where name
='customer_c1')
    drop index customer.customer_c1

if exists (select name from sysindexes where name
='history_c1')
    drop index history.history_c1
go

create unique clustered index customer_c1 on
customer(c_w_id, c_d_id, c_id)
with SORT_IN_TEMPDB, fillfactor=100 on cs_fg
go

create unique nonclustered index customer_nc1 on
customer(c_w_id, c_d_id, c_last, c_first, c_id)
with SORT_IN_TEMPDB, fillfactor=100
go

create clustered index history_c1 on history(h_c_w_id,
h_date, h_c_d_id, h_c_id, h_amount)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

--
-- 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

--
-- HPTPCC Setup Kit
-- Create ORDERS Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go

if exists ( select name from sysindexes where name =
'orders_c1' )
    drop index orders.orders_c1

if exists ( select name from sysindexes where name =
'order_line_c1' )
    drop index order_line.order_line_c1

if exists ( select name from sysindexes where name =
'new_order_c1' )
    drop index new_order.new_order_c1
go

create unique clustered index orders_c1 on
orders(o_w_id, o_d_id, o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

--create index orders_nc1 on orders(o_w_id, o_d_id,
o_c_id, o_id)
--with SORT_IN_TEMPDB, fillfactor=100
--go

create unique clustered index new_order_c1 on
new_order(no_w_id, no_d_id, no_o_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index order_line_c1 on
order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

--
-- HPTPCC Setup Kit
-- Create WAREHOUSE Indexes
-- and move the table from LoadFg to MiscFg
--
use tpcc
go

if exists (select name from sysindexes where name
='stock_c1')
    drop index stock.stock_c1
go

create unique clustered index warehouse_c1 on
warehouse(w_id)
with SORT_IN_TEMPDB, fillfactor=100 on misc_fg
go

create unique clustered index district_c1 on
district(d_w_id, d_id)
with SORT_IN_TEMPDB, fillfactor=100 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

```

Non Clustered Index Creation Scripts

Incompassed in previous section

B.3 Stored Procedures

CreateNewOrdProc.sql

```

-----
-- File: NEWORD.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
--
-- Copyright Microsoft, 2005 --
--
-- Creates neworder stored procedure
--
-- Interface Level: 4.20.000 --
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_neworder' )
    DROP PROCEDURE tpcc_neworder
GO

CREATE PROCEDURE tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0,
@ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0,
@ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0,
@ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0,
@ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0,
@ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0,
@ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0,
@ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0,
@ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0,
@ol_qty9 smallint = 0,

```

```

        @i_id10 int = 0, @s_w_id10 int = 0,
@o_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11 int = 0,
@o_qty11 smallint = 0,
        @i_id12 int = 0, @s_w_id12 int = 0,
@o_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13 int = 0,
@o_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14 int = 0,
@o_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15 int = 0,
@o_qty15 smallint = 0

```

AS

```

DECLARE @w_tax      smallmoney,
        @d_tax      smallmoney,
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount smallmoney,
        @i_price    smallmoney,
        @i_name     char(24),
        @i_data     char(50),
        @o_entry_d  datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data     char(50),
        @s_dist     char(24),
        @li_no      int,
        @o_id       int,
        @commit_flag tinyint,
        @li_id      int,
        @li_s_w_id  int,
        @li_qty     smallint,
        @o_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_o_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 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 @o_qty1
           WHEN 2 THEN @o_qty2
           WHEN 3 THEN @o_qty3
           WHEN 4 THEN @o_qty4
           WHEN 5 THEN @o_qty5
           WHEN 6 THEN @o_qty6
           WHEN 7 THEN @o_qty7
           WHEN 8 THEN @o_qty8
           WHEN 9 THEN @o_qty9
           WHEN 10 THEN @o_qty10
           WHEN 11 THEN @o_qty11
           WHEN 12 THEN @o_qty12
           WHEN 13 THEN @o_qty13
           WHEN 14 THEN @o_qty14
           WHEN 15 THEN @o_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 9 THEN s_dist_09
        WHEN 9 THEN s_dist_09
        WHEN 10 THEN s_dist_10
    END

```

```

WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id

```

```

--
-- if there actually is a stock (and item) with these ids,
go to work

```

```

IF (@@rowcount > 0)
BEGIN

```

-- insert order_line data (using data from item and stock)

```

INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)

```

-- send line-item data to client

```

SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

```

(patindex('%ORIGINAL%',@s_data) > 0))

```

        THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
    END
    ELSE
    BEGIN

```

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

```

SELECT ",0",0,0
SELECT @commit_flag = 0
    END

```

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

```

SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM   customer WITH (repeatableread)
WHERE  c_id     = @c_id AND
       c_w_id   = @w_id AND
       c_d_id   = @d_id

```

-- insert fresh row into orders table

```

INSERT INTO orders VALUES ( @o_id,
                             @d_id,
                             @w_id,
                             @c_id_local,

```

```

0,
@o_ol_cnt,
@o_all_local,
@o_entry_d)
-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                                @d_id,
                                @w_id)
-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)
    COMMIT TRANSACTION n
ELSE
-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n
-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @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 PROCEDURE sp0tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0,
@o_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0,
@o_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0,
@o_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0,
@o_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0,
@o_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0,
@o_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0,
@o_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0,
@o_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0,
@o_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0,
@o_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0,
@o_qty11 smallint = 0,

```

```

    @i_id12 int = 0, @s_w_id12 int = 0,
@o_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0,
@o_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0,
@o_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0,
@o_qty15 smallint = 0
AS
DECLARE @w_tax smallmoney,
        @d_tax smallmoney,
        @c_last char(16),
        @c_credit char(2),
        @c_discount smallmoney,
        @i_price smallmoney,
        @i_name char(24),
        @i_data char(50),
        @o_entry_d datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data char(50),
        @s_dist char(24),
        @li_no int,
        @o_id int,
        @commit_flag tinyint,
        @li_id int,
        @li_s_w_id int,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int
BEGIN
BEGIN TRANSACTION n
-----
-- get district tax and next 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 - @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
    WHEN 11 THEN s_dist_11
    WHEN 12 THEN s_dist_12

```

```

WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id
-----
-- if there actually is a stock (and item) with these ids,
go to work
-----
IF (@@rowcount > 0)
BEGIN
-----
-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES( @o_id,
                               @d_id,
                               @w_id,
                               @li_no,
                               @li_id,
                               'dec 31, 1899',
                               @i_price * @li_qty,
                               @li_s_w_id,
                               @li_qty,
                               @s_dist)
-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
END
ELSE
BEGIN
-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
END
END
-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id
-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                            @d_id,
                            @w_id,
                            @c_id_local,
                            0,
                            @o_ol_cnt,
                            @o_all_local,
                            @o_entry_d)

```

```

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                               @d_id,
                               @w_id)
-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

COMMIT TRANSACTION n

ELSE
-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n
-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag
END
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp1tpcc_neworder' )
DROP PROCEDURE sp1tpcc_neworder
GO
CREATE PROCEDURE sp1tpcc_neworder
@w_id int,
@d_id tinyint,
@c_id int,
@o_ol_cnt tinyint,
@o_all_local tinyint,
@i_id1 int = 0, @s_w_id1 int = 0,
@ol_qty1 smallint = 0,
@i_id2 int = 0, @s_w_id2 int = 0,
@ol_qty2 smallint = 0,
@i_id3 int = 0, @s_w_id3 int = 0,
@ol_qty3 smallint = 0,
@i_id4 int = 0, @s_w_id4 int = 0,
@ol_qty4 smallint = 0,
@i_id5 int = 0, @s_w_id5 int = 0,
@ol_qty5 smallint = 0,
@i_id6 int = 0, @s_w_id6 int = 0,
@ol_qty6 smallint = 0,
@i_id7 int = 0, @s_w_id7 int = 0,
@ol_qty7 smallint = 0,
@i_id8 int = 0, @s_w_id8 int = 0,
@ol_qty8 smallint = 0,
@i_id9 int = 0, @s_w_id9 int = 0,
@ol_qty9 smallint = 0,
@i_id10 int = 0, @s_w_id10 int = 0,
@ol_qty10 smallint = 0,
@i_id11 int = 0, @s_w_id11 int = 0,
@ol_qty11 smallint = 0,
@i_id12 int = 0, @s_w_id12 int = 0,
@ol_qty12 smallint = 0,
@i_id13 int = 0, @s_w_id13 int = 0,
@ol_qty13 smallint = 0,
@i_id14 int = 0, @s_w_id14 int = 0,
@ol_qty14 smallint = 0,

```

```

@i_id15 int = 0, @s_w_id15 int = 0,
@ol_qty15 smallint = 0
AS
DECLARE @w_tax smallmoney,
        @d_tax smallmoney,
        @c_last char(16),
        @c_credit char(2),
        @c_discount smallmoney,
        @i_price smallmoney,
        @i_name char(24),
        @i_data char(50),
        @o_entry_d datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data char(50),
        @s_dist char(24),
        @li_no int,
        @o_id int,
        @commit_flag tinyint,
        @li_id int,
        @li_s_w_id int,
        @li_qty smallint,
        @ol_number int,
        @c_id_local int
BEGIN
BEGIN TRANSACTION n
-----
-- get district tax and next 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 @i_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 = @i_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 @d_id
WHEN 1 THEN s_dist_01
WHEN 2 THEN s_dist_02
WHEN 3 THEN s_dist_03
WHEN 4 THEN s_dist_04
WHEN 5 THEN s_dist_05
WHEN 6 THEN s_dist_06
WHEN 7 THEN s_dist_07
WHEN 8 THEN s_dist_08
WHEN 9 THEN s_dist_09
WHEN 10 THEN s_dist_10
END
WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id
-----

```

```

-- if there actually is a stock (and item) with these ids,
go to work
-----
IF (@@rowcount > 0)
BEGIN
-----
-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)
-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND
(patindex('%ORIGINAL%',@s_data) > 0) )
THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
END
ELSE
BEGIN
-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
END
-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id
-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                             @d_id,
                             @w_id,
                             @c_id_local,
                             0,
                             @o_ol_cnt,
                             @o_all_local,
                             @o_entry_d)
-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                                @d_id,

```

```

                                @w_id)
-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

COMMIT TRANSACTION n
ELSE
-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n
-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @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 PROCEDURE sp2tpcc_neworder
@w_id int,
@d_id tinyint,
@c_id int,
@o_ol_cnt tinyint,
@o_all_local tinyint,
@i_id1 int = 0, @s_w_id1 int = 0,
@ol_qty1 smallint = 0,
@i_id2 int = 0, @s_w_id2 int = 0,
@ol_qty2 smallint = 0,
@i_id3 int = 0, @s_w_id3 int = 0,
@ol_qty3 smallint = 0,
@i_id4 int = 0, @s_w_id4 int = 0,
@ol_qty4 smallint = 0,
@i_id5 int = 0, @s_w_id5 int = 0,
@ol_qty5 smallint = 0,
@i_id6 int = 0, @s_w_id6 int = 0,
@ol_qty6 smallint = 0,
@i_id7 int = 0, @s_w_id7 int = 0,
@ol_qty7 smallint = 0,
@i_id8 int = 0, @s_w_id8 int = 0,
@ol_qty8 smallint = 0,
@i_id9 int = 0, @s_w_id9 int = 0,
@ol_qty9 smallint = 0,
@i_id10 int = 0, @s_w_id10 int = 0,
@ol_qty10 smallint = 0,
@i_id11 int = 0, @s_w_id11 int = 0,
@ol_qty11 smallint = 0,
@i_id12 int = 0, @s_w_id12 int = 0,
@ol_qty12 smallint = 0,
@i_id13 int = 0, @s_w_id13 int = 0,
@ol_qty13 smallint = 0,
@i_id14 int = 0, @s_w_id14 int = 0,
@ol_qty14 smallint = 0,
@i_id15 int = 0, @s_w_id15 int = 0,
@ol_qty15 smallint = 0

```

```

AS
DECLARE @w_tax      smallmoney,
        @d_tax      smallmoney,
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount smallmoney,
        @i_price    smallmoney,
        @i_name     char(24),
        @i_data     char(50),
        @o_entry_d  datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data     char(50),
        @s_dist     char(24),
        @li_no      int,
        @o_id       int,
        @commit_flag tinyint,
        @li_id      int,
        @li_s_w_id  int,
        @li_qty     smallint,
        @ol_number  int,
        @c_id_local int

BEGIN

BEGIN TRANSACTION n

-----
-- get district tax and next 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 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 (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 = s_remote_cnt +
           CASE WHEN (@li_s_w_id =
                      @w_id) THEN 0 ELSE 1 END,
         @s_data     = s_data,
         @s_dist     = CASE @d_id
                       WHEN 1 THEN s_dist_01
                       WHEN 2 THEN s_dist_02
                       WHEN 3 THEN s_dist_03
                       WHEN 4 THEN s_dist_04
                       WHEN 5 THEN s_dist_05
                       WHEN 6 THEN s_dist_06
                       WHEN 7 THEN s_dist_07
                       WHEN 8 THEN s_dist_08
                       WHEN 9 THEN s_dist_09
                       WHEN 10 THEN s_dist_10
                       END
         WHERE s_i_id = @li_id AND
              s_w_id = @li_s_w_id

-----
--
-- if there actually is a stock (and item) with these ids,
go to work
-----
--
-- IF (@@rowcount > 0)

```

```

-----BEGIN-----
-- insert order_line data (using data from item and
stock)
-----
    INSERT INTO order_line VALUES( @o_id,
                                   @d_id,
                                   @w_id,
                                   @li_no,
                                   @li_id,
                                   'dec 31, 1899',
                                   @i_price * @li_qty,
                                   @li_s_w_id,
                                   @li_qty,
                                   @s_dist)

-----
-- send line-item data to client
-----
    SELECT @i_name,
           @s_quantity,
           b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )
           THEN 'B' ELSE 'G' END,
           @i_price,
           @i_price * @li_qty
           END
           ELSE
           BEGIN

-----
-- no item (or stock) found - triggers rollback
condition
-----
    SELECT ",0",0,0
    SELECT @commit_flag = 0
           END

-----
-- get customer last name, discount, and credit rating
-----
    SELECT @c_last = c_last,
           @c_discount = c_discount,
           @c_credit = c_credit,
           @c_id_local = c_id
    FROM   customer WITH (repeatable)
    WHERE  c_id     = @c_id AND
           c_w_id   = @w_id AND
           c_d_id   = @d_id

-----
-- insert fresh row into orders table
-----
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
    INSERT INTO new_order VALUES ( @o_id,
                                    @d_id,
                                    @w_id)

-----
-- select warehouse tax
-----

```

```

SELECT @w_tax = w_tax
FROM warehouse WITH (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 = 'sp3tpcc_neworder' )
    DROP PROCEDURE sp3tpcc_neworder
GO

CREATE PROCEDURE sp3tpcc_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,
    @o_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0,
    @o_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0,
    @o_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0,
    @o_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0,
    @o_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0,
    @o_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0,
    @o_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0,
    @o_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0,
    @o_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0,
    @o_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0,
    @o_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0,
    @o_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0,
    @o_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0,
    @o_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0,
    @o_qty15 smallint = 0

AS
DECLARE @w_tax smallmoney,
       @d_tax smallmoney,
       @c_last char(16),

```

```

@c_discount smallmoney,
@i_price smallmoney,
@i_name char(24),
@i_data char(50),
@o_entry_d datetime,
@remote_flag int,
@s_quantity smallint,
@s_data char(50),
@s_dist char(24),
@li_no int,
@o_id int,
@commit_flag tinyint,
@li_id int,
@li_s_w_id int,
@li_qty smallint,
@ol_number int,
@c_id_local int

BEGIN
BEGIN TRANSACTION n
-- get district tax and next 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_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 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 (repeatable)
WHERE i_id = @li_id
-- update stock values
UPDATE stock
SET s_ytd = s_ytd + @li_qty,
    @i_qty + @s_quantity = s_quantity
CASE WHEN (s_quantity -
@li_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt = s_order_cnt + 1,
    s_remote_cnt = s_remote_cnt +
CASE WHEN (@li_s_w_id =
@w_id) THEN 0 ELSE 1 END,
    @s_data = s_data,
    @s_dist = CASE @d_id
    WHEN 1 THEN s_dist_01
    WHEN 2 THEN s_dist_02
    WHEN 3 THEN s_dist_03
    WHEN 4 THEN s_dist_04
    WHEN 5 THEN s_dist_05
    WHEN 6 THEN s_dist_06
    WHEN 7 THEN s_dist_07
    WHEN 8 THEN s_dist_08
    WHEN 9 THEN s_dist_09
    WHEN 10 THEN s_dist_10
END
WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id
-- if there actually is a stock (and item) with these ids,
go to work
IF (@@rowcount > 0)
BEGIN
-- insert order_line data (using data from item and
stock)

```

```

INSERT INTO order_line VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)

-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )

       THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
END
ELSE
BEGIN
-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
END
END

-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                             @d_id,
                             @w_id,
                             @c_id_local,
                             0,
                             @o_ol_cnt,
                             @o_all_local,
                             @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                                @d_id,
                                @w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

```

```

IF (@commit_flag = 1)
COMMIT TRANSACTION n
ELSE
-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n

-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag
END
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp4tpcc_neworder' )
DROP PROCEDURE sp4tpcc_neworder
GO

CREATE PROCEDURE sp4tpcc_neworder
@w_id int,
@d_id tinyint,
@c_id int,
@o_ol_cnt tinyint,
@o_all_local tinyint,
@i_id1 int = 0, @s_w_id1 int = 0,
@o_qty1 smallint = 0,
@i_id2 int = 0, @s_w_id2 int = 0,
@o_qty2 smallint = 0,
@i_id3 int = 0, @s_w_id3 int = 0,
@o_qty3 smallint = 0,
@i_id4 int = 0, @s_w_id4 int = 0,
@o_qty4 smallint = 0,
@i_id5 int = 0, @s_w_id5 int = 0,
@o_qty5 smallint = 0,
@i_id6 int = 0, @s_w_id6 int = 0,
@o_qty6 smallint = 0,
@i_id7 int = 0, @s_w_id7 int = 0,
@o_qty7 smallint = 0,
@i_id8 int = 0, @s_w_id8 int = 0,
@o_qty8 smallint = 0,
@i_id9 int = 0, @s_w_id9 int = 0,
@o_qty9 smallint = 0,
@i_id10 int = 0, @s_w_id10 int = 0,
@o_qty10 smallint = 0,
@i_id11 int = 0, @s_w_id11 int = 0,
@o_qty11 smallint = 0,
@i_id12 int = 0, @s_w_id12 int = 0,
@o_qty12 smallint = 0,
@i_id13 int = 0, @s_w_id13 int = 0,
@o_qty13 smallint = 0,
@i_id14 int = 0, @s_w_id14 int = 0,
@o_qty14 smallint = 0,
@i_id15 int = 0, @s_w_id15 int = 0,
@o_qty15 smallint = 0

AS
DECLARE @w_tax smallmoney,
        @d_tax smallmoney,
        @c_last char(16),
        @c_credit char(2),
        @c_discount smallmoney,

```

```

@i_price smallmoney,
@d_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     = @i_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 @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,

```

```

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

-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )
        THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
    END
    ELSE
    BEGIN

-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
    END

-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM   customer WITH (repeatableread)
WHERE  c_id     = @c_id AND
       c_w_id   = @w_id AND
       c_d_id   = @d_id

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
       @d_id,
       @w_id,
       @c_id_local,
       0,
       @o_ol_cnt,
       @o_all_local,
       @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
       @d_id,
       @w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM   warehouse WITH (repeatableread)
WHERE  w_id   = @w_id

IF (@commit_flag = 1)

```

```

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 = 'sp5tpcc_neworder' )
    DROP PROCEDURE sp5tpcc_neworder
GO

CREATE PROCEDURE sp5tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0,
    @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0,
    @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0,
    @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0,
    @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0,
    @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0,
    @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0,
    @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0,
    @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0,
    @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0,
    @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0,
    @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0,
    @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0,
    @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0,
    @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0,
    @ol_qty15 smallint = 0

AS
DECLARE @w_tax smallmoney,
       @d_tax smallmoney,
       @c_last char(16),
       @c_credit char(2),
       @c_discount smallmoney,
       @i_price smallmoney,
       @i_name char(24),
       @i_data char(50),
       @o_entry_d datetime,
       @remote_flag int,

```

```

@s_quantity smallint,
@s_data char(50),
@s_dist char(24),
@li_no int,
@o_id int,
@commit_flag tinyint,
@li_id int,
@li_s_w_id int,
@li_qty smallint,
@ol_number int,
@c_id_local int

BEGIN

BEGIN TRANSACTION n

-----
-- get district tax and next 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 - @li_qty +
    CASE WHEN (s_quantity -
@li_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt = s_order_cnt + 1,
    s_remote_cnt = s_remote_cnt +
    CASE WHEN (@li_s_w_id =
@w_id) THEN 0 ELSE 1 END,
    @s_data = s_data,
    @s_dist = CASE @d_id
        WHEN 1 THEN s_dist_01
        WHEN 2 THEN s_dist_02
        WHEN 3 THEN s_dist_03
        WHEN 4 THEN s_dist_04
        WHEN 5 THEN s_dist_05
        WHEN 6 THEN s_dist_06
        WHEN 7 THEN s_dist_07
        WHEN 8 THEN s_dist_08
        WHEN 9 THEN s_dist_09
        WHEN 10 THEN s_dist_10
    END
WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id

-----
-- if there actually is a stock (and item) with these ids,
go to work
-----
--
-- IF (@@rowcount > 0)
-- BEGIN
-----
-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES( @o_id,
    @d_id,
    @w_id,
    @li_no,
    @li_id,
    'dec 31, 1899',

```

```

    @li_price_id @li_qty,
    @li_qty,
    @s_dist)

-----
-- send line-item data to client
-----
SELECT @i_name,
    @s_quantity,
    b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )

    THEN 'B' ELSE 'G' END,
    @i_price,
    @i_price * @li_qty
    END
    ELSE
    BEGIN

-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",",0,0,0
SELECT @commit_flag = 0
    END

-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
    @c_discount = c_discount,
    @c_credit = c_credit,
    @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
    @d_id,
    @w_id,
    @c_id_local,
    0,
    @o_ol_cnt,
    @o_all_local,
    @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
    @d_id,
    @w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

    COMMIT TRANSACTION n
ELSE

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

```

```

ROLLBACK TRANSACTION n
-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @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 PROCEDURE sp6tpcc_neworder
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0,
    @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0,
    @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0,
    @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0,
    @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0,
    @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0,
    @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0,
    @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0,
    @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0,
    @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0,
    @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0,
    @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0,
    @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0,
    @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0,
    @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0,
    @ol_qty15 smallint = 0

AS
DECLARE @w_tax smallmoney,
        @d_tax smallmoney,
        @c_last char(16),
        @c_credit char(2),
        @c_discount smallmoney,
        @i_price smallmoney,
        @i_name char(24),
        @i_data char(50),
        @o_entry_d datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data char(50),
        @s_dist char(24),
        @li_no int,

```

```

        @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 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 +
- @li_qty +
CASE WHEN (s_quantity -
@li_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt = s_order_cnt + 1,
    s_remote_cnt = s_remote_cnt +
CASE WHEN (@li_s_w_id =
@w_id) THEN 0 ELSE 1 END,
    @s_data = s_data,
    @s_dist = CASE @d_id
    WHEN 1 THEN s_dist_01
    WHEN 2 THEN s_dist_02
    WHEN 3 THEN s_dist_03
    WHEN 4 THEN s_dist_04
    WHEN 5 THEN s_dist_05
    WHEN 6 THEN s_dist_06
    WHEN 7 THEN s_dist_07
    WHEN 8 THEN s_dist_08
    WHEN 9 THEN s_dist_09
    WHEN 10 THEN s_dist_10
END
WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id

-----
-- if there actually is a stock (and item) with these ids,
go to work
-----
IF (@@rowcount > 0)
BEGIN

-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                @li_qty,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)

```

```

-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN (
(patindex('%ORIGINAL%',@i_data) > 0) AND

(patindex('%ORIGINAL%',@s_data) > 0) )

THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
END
ELSE
BEGIN
-----
-- no item (or stock) found - triggers rollback
condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
END
-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id
-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                            @d_id,
                            @w_id,
                            @c_id_local,
                            0,
                            @o_ol_cnt,
                            @o_all_local,
                            @o_entry_d)
-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                               @d_id,
                               @w_id)
-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

COMMIT TRANSACTION n

ELSE

-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n
-----
-- return order data to client

```

```

-----
SELECT @w_tax,-----
       @d_tax,
       @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 PROCEDURE sp7tpcc_neworder
       @w_id int,
       @d_id tinyint,
       @c_id int,
       @o_ol_cnt tinyint,
       @o_all_local tinyint,
       @i_id1 int = 0, @s_w_id1 int = 0,
@ol_qty1 smallint = 0,
       @i_id2 int = 0, @s_w_id2 int = 0,
@ol_qty2 smallint = 0,
       @i_id3 int = 0, @s_w_id3 int = 0,
@ol_qty3 smallint = 0,
       @i_id4 int = 0, @s_w_id4 int = 0,
@ol_qty4 smallint = 0,
       @i_id5 int = 0, @s_w_id5 int = 0,
@ol_qty5 smallint = 0,
       @i_id6 int = 0, @s_w_id6 int = 0,
@ol_qty6 smallint = 0,
       @i_id7 int = 0, @s_w_id7 int = 0,
@ol_qty7 smallint = 0,
       @i_id8 int = 0, @s_w_id8 int = 0,
@ol_qty8 smallint = 0,
       @i_id9 int = 0, @s_w_id9 int = 0,
@ol_qty9 smallint = 0,
       @i_id10 int = 0, @s_w_id10 int = 0,
@ol_qty10 smallint = 0,
       @i_id11 int = 0, @s_w_id11 int = 0,
@ol_qty11 smallint = 0,
       @i_id12 int = 0, @s_w_id12 int = 0,
@ol_qty12 smallint = 0,
       @i_id13 int = 0, @s_w_id13 int = 0,
@ol_qty13 smallint = 0,
       @i_id14 int = 0, @s_w_id14 int = 0,
@ol_qty14 smallint = 0,
       @i_id15 int = 0, @s_w_id15 int = 0,
@ol_qty15 smallint = 0

AS
DECLARE @w_tax smallmoney,
       @d_tax smallmoney,
       @c_last char(16),
       @c_credit char(2),
       @c_discount smallmoney,
       @i_price smallmoney,
       @i_name char(24),
       @i_data char(50),
       @o_entry_d datetime,
       @remote_flag int,
       @s_quantity smallint,
       @s_data char(50),
       @s_dist char(24),
       @li_no int,
       @o_id int,
       @commit_flag tinyint,
       @li_id int,
       @li_s_w_id int,
       @li_qty smallint,

```

```

@li_id_local int,
@li_no 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 (repeatable)
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 @d_id
WHEN 1 THEN s_dist_01
WHEN 2 THEN s_dist_02
WHEN 3 THEN s_dist_03
WHEN 4 THEN s_dist_04
WHEN 5 THEN s_dist_05
WHEN 6 THEN s_dist_06
WHEN 7 THEN s_dist_07
WHEN 8 THEN s_dist_08
WHEN 9 THEN s_dist_09
WHEN 10 THEN s_dist_10
END
WHERE s_i_id = @li_id AND
      s_w_id = @li_s_w_id
-----
-- if there actually is a stock (and item) with these ids,
go to work
-----
IF (@@rowcount > 0)
BEGIN
-----
-- insert order_line data (using data from item and
stock)
-----
INSERT INTO order_line VALUES( @o_id,
                               @d_id,
                               @w_id,
                               @li_no,
                               @li_id,
                               'dec 31, 1899',
                               @i_price * @li_qty,
                               @li_s_w_id,
                               @li_qty,
                               @s_dist)
-----
-- send line-item data to client
-----
SELECT @i_name,

```

```

@o_id, @c_id, 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 (repeatable)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id
-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
                             @d_id,
                             @w_id,
                             @c_id_local,
                             0,
                             @o_ol_cnt,
                             @o_all_local,
                             @o_entry_d)
-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                                @d_id,
                                @w_id)
-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatable)
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_credit,
@o_entry_d,
@commit_flag
END
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
-----
tpcc_neworder_new.sql
-----
-- File: TPCC_NEWORDER_NEW.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
-- This acid stored procedure implements the
neworder --
-- transaction. It outputs timestamps at the
-- beginning of the transaction, before the
commit --
-- delay, and after the commit.
-----
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO
USE tpcc
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_neworder_new' )
DROP PROCEDURE tpcc_neworder_new
GO
-- neworder_new v2.5 6/23/05 PeterCa
-- 1q stock/order_line/client. upd district & ins
neworder.
-- cust/warehouse select together, ins order separate
-- uses rownumber to distinct w any transform
-- uses in-memory sort for distinct on iid,wid
-- uses charindex
-- will rollback if (@i_idX,@s_w_idX pairs not unique)
OR (@i_idX not unique).
CREATE PROCEDURE tpcc_neworder_new
@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,
@o_qty3 smallint = 0,
        @i_id4 int = 0, @s_w_id4 int = 0,
@o_qty4 smallint = 0,
        @i_id5 int = 0, @s_w_id5 int = 0,
@o_qty5 smallint = 0,
        @i_id6 int = 0, @s_w_id6 int = 0,
@o_qty6 smallint = 0,
        @i_id7 int = 0, @s_w_id7 int = 0,
@o_qty7 smallint = 0,
        @i_id8 int = 0, @s_w_id8 int = 0,
@o_qty8 smallint = 0,
        @i_id9 int = 0, @s_w_id9 int = 0,
@o_qty9 smallint = 0,
        @i_id10 int = 0, @s_w_id10 int = 0,
@o_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11 int = 0,
@o_qty11 smallint = 0,
        @i_id12 int = 0, @s_w_id12 int = 0,
@o_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13 int = 0,
@o_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14 int = 0,
@o_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15 int = 0,
@o_qty15 smallint = 0

AS
BEGIN
DECLARE @o_id int,
        @d_tax smallmoney,
        @o_entry_d datetime,
        @commit_flag tinyint

BEGIN TRANSACTION n
-- get district tax and next available order id and
update
-- insert corresponding row into new-order table
-- plus initialize local variables

UPDATE district
SET @d_tax = d_tax,
    @o_id = d_next_o_id,
    d_next_o_id = d_next_o_id + 1,
    @o_entry_d = GETDATE(),
    @commit_flag = 1
OUTPUT deleted.d_next_o_id,
        @d_id,
        @w_id
INTO new_order
WHERE d_w_id = @w_id AND
    d_id = @d_id

-- update stock from stock join (item join (params))
-- output to orderline, output to client
-- NOTE: @@rowcount != @o_o_cnt
-- if (@i_idX,@s_w_idX pairs not unique)
OR (@i_idX not unique).

UPDATE stock
SET s_ytd = s_ytd + info.ol_qty,
    s_quantity = s_quantity - info.ol_qty +
CASE WHEN (s_quantity -
info.ol_qty < 10) THEN 91 ELSE 0 END,
    s_order_cnt = s_order_cnt + 1,
    s_remote_cnt = s_remote_cnt +

CASE WHEN (info.w_id = @w_id) THEN 0 ELSE 1 END

OUTPUT @o_id,
        @d_id,
        @w_id,
        info.lino,

```

```

        info.i_name, 1899",
        info.i_price * info.ol_qty,
        info.w_id,
        info.ol_qty,
CASE @d_id WHEN 1 THEN
inserted.s_dist_01
        WHEN 2 THEN inserted.s_dist_02
        WHEN 3 THEN inserted.s_dist_03
        WHEN 4 THEN inserted.s_dist_04
        WHEN 5 THEN inserted.s_dist_05
        WHEN 6 THEN inserted.s_dist_06
        WHEN 7 THEN inserted.s_dist_07
        WHEN 8 THEN inserted.s_dist_08
        WHEN 9 THEN inserted.s_dist_09
        WHEN 10 THEN inserted.s_dist_10

END
INTO order_line

OUTPUT info.i_name,inserted.s_quantity,
CASE WHEN
((charindex("ORIGINAL",info.i_data) > 0) AND
(charindex("ORIGINAL",inserted.s_data) > 0) )
THEN "B" ELSE "G" END,
        info.i_price,
        info.i_price*info.ol_qty
FROM stock INNER JOIN
(SELECT iid,
        wid,
        lino,
        ol_qty,
        i_price,
        i_name,
        i_data
FROM (SELECT iid,
        wid,
        lino,
        qty,
row_number() OVER (PARTITION BY iid,wid ORDER
BY iid,wid)
FROM (SELECT
        @i_id1,@s_w_id1,1,@o_qty1 UNION ALL
SELECT
        @i_id2,@s_w_id2,2,@o_qty2 UNION ALL
SELECT
        @i_id3,@s_w_id3,3,@o_qty3 UNION ALL
SELECT
        @i_id4,@s_w_id4,4,@o_qty4 UNION ALL
SELECT
        @i_id5,@s_w_id5,5,@o_qty5 UNION ALL
SELECT
        @i_id6,@s_w_id6,6,@o_qty6 UNION ALL
SELECT
        @i_id7,@s_w_id7,7,@o_qty7 UNION ALL
SELECT
        @i_id8,@s_w_id8,8,@o_qty8 UNION ALL
SELECT
        @i_id9,@s_w_id9,9,@o_qty9 UNION ALL
SELECT
        @i_id10,@s_w_id10,10,@o_qty10 UNION ALL
SELECT
        @i_id11,@s_w_id11,11,@o_qty11 UNION ALL
SELECT
        @i_id12,@s_w_id12,12,@o_qty12 UNION ALL
SELECT
        @i_id13,@s_w_id13,13,@o_qty13 UNION ALL
SELECT
        @i_id14,@s_w_id14,14,@o_qty14 UNION ALL
SELECT
        @i_id15,@s_w_id15,15,@o_qty15) AS
uo1(iid,wid,lino,qty)
        ) AS
o1(iid,wid,lino,ol_qty,rownum)
INNER JOIN

```

```

        item (repeatable) ON i_id =
iid AND -- filters out invalid items rownum = 1
) AS
info(i_id,w_id,lino,ol_qty,i_price,i_name,i_data)
ON s_i_id = info.i_id AND
    s_w_id = info.w_id

IF (@@rowcount <> @o_o_cnt) -- must have an
invalid item
SELECT @commit_flag = 0 -- 2.4.2.3 requires
rest to proceed

-- insert fresh row into orders table
INSERT INTO orders VALUES ( @o_id,
        @d_id,
        @w_id,
        @c_id,
        0,
        @o_o_cnt,
        @o_all_local,
        @o_entry_d)

-- get customer last name, discount, and credit
rating
-- get warehouse tax
-- return order_data to client
SELECT w_tax,
        @d_tax,
        @o_id,
        c_last,
        c_discount,
        c_credit,
        @o_entry_d,
        @commit_flag
FROM warehouse(repeatable),
customer(repeatable)
WHERE w_id = @w_id AND
    c_id = @c_id AND
    c_w_id = @w_id AND
    c_d_id = @d_id

-- @@rowcount checks that previous
select found a valid customer
IF (@@rowcount = 0)
BEGIN
        RAISERROR('Invalid
Customer ID', 11, 1 )
ROLLBACK TRANSACTION n
END
ELSE IF (@commit_flag = 1)
COMMIT TRANSACTION n
ELSE -- all that work for nothing.
ROLLBACK TRANSACTION n

END
GO

CreateOrdStatProc.sql

-----
-- File: ORDSTAT.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.68 --
-- Copyright Microsoft, 2006 --
--

```

```

--      Creates order status stored procedure
--
--
--      Interface Level: 4.20.000      --
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_orderstatus' )
DROP PROCEDURE tpcc_orderstatus
GO

CREATE PROCEDURE tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT @cnt = (count(*)+1)/2
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id = @w_id AND
c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,
        @c_balance = c_balance,
        @c_first = c_first,
        @c_last = c_last,
        @c_middle = c_middle
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id = @w_id AND
c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT @c_balance = c_balance,
        @c_first = c_first,
        @c_middle = c_middle,
        @c_last = c_last
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND

```

```

        @w_id = @w_id AND
        @d_id = @d_id AND
        @o_id = @o_id AND
        @o_entry_d = @o_entry_d AND
        @o_carrier_id = @o_carrier_id AND
        @c_balance = @c_balance AND
        @c_first = @c_first AND
        @c_last = @c_last AND
        @c_middle = @c_middle AND
        @cnt = @cnt
COMMIT TRANSACTION o
END
-----
-- 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 PROCEDURE sp0tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,

```

```

        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT @cnt = (count(*)+1)/2
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id = @w_id AND
c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,
        @c_balance = c_balance,
        @c_first = c_first,
        @c_last = c_last,
        @c_middle = c_middle
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id = @w_id AND
c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT @c_balance = c_balance,
        @c_first = c_first,
        @c_middle = c_middle,
        @c_last = c_last
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
c_d_id = @d_id AND
c_w_id = @w_id

SET @cnt = @@rowcount
END
-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
RAISERROR('Customer not found',18,1)
GOTO custnotfound
END
-----
-- get order info
-----
SELECT @o_id = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
FROM orders WITH (serializable)
WHERE o_c_id = @c_id AND
o_d_id = @d_id AND
o_w_id = @w_id
ORDER BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM order_line WITH (repeatableread)
WHERE ol_o_id = @o_id AND
ol_d_id = @d_id AND
ol_w_id = @w_id

custnotfound:
COMMIT TRANSACTION o

```

```

        ol_delivery_d
FROM   order_line WITH (repeatableread)
WHERE  ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,

        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp1tpcc_orderstatus' )
DROP PROCEDURE sp1tpcc_orderstatus

CREATE PROCEDURE sp1tpcc_orderstatus
        @w_id    int,
        @d_id    tinyint,

        @c_id    int,
        @c_last  char(16) = ""

AS
DECLARE @c_balance  money,
        @c_first   char(16),
        @c_middle  char(2),
        @o_id     int,
        @o_entry_d  datetime,
        @o_carrier_id smallint,
        @cnt      smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT  @cnt = (count(*)+1)/2
FROM    customer WITH (repeatableread)
WHERE   c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

SET     rowcount @cnt

SELECT  @c_id   = c_id,
        @c_balance = c_balance,
        @c_first  = c_first,
        @c_last   = c_last,
        @c_middle = c_middle
FROM    customer WITH (repeatableread)
WHERE   c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

ORDER  BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get order info
-----
SELECT  @o_id      = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id = @c_id AND
       o_d_id = @d_id AND
       o_w_id = @w_id

ORDER  BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,

        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp2tpcc_orderstatus' )
DROP PROCEDURE sp2tpcc_orderstatus

CREATE PROCEDURE sp2tpcc_orderstatus
        @w_id    int,

```

```

BEGIN-----
-- get customer info if by id
-----
SELECT  @c_balance = c_balance,
        @c_first   = c_first,
        @c_middle  = c_middle,
        @c_last    = c_last
FROM    customer WITH (repeatableread)
WHERE   c_id = @c_id AND
       c_d_id = @d_id AND
       c_w_id = @w_id

SELECT  @cnt      = @@rowcount
END

-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
RAISERROR('Customer not found',18,1)
GOTO custnotfound
END

-----
-- get order info
-----
SELECT  @o_id      = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id = @c_id AND
       o_d_id = @d_id AND
       o_w_id = @w_id

ORDER  BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line WITH (repeatableread)
WHERE   ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,

        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp2tpcc_orderstatus' )
DROP PROCEDURE sp2tpcc_orderstatus

CREATE PROCEDURE sp2tpcc_orderstatus
        @w_id    int,

```

```

        @d_id    tinyint,

        @c_id    int,
        @c_last  char(16) = ""

AS
DECLARE @c_balance  money,
        @c_first   char(16),
        @c_middle  char(2),
        @o_id     int,
        @o_entry_d  datetime,
        @o_carrier_id smallint,
        @cnt      smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT  @cnt = (count(*)+1)/2
FROM    customer WITH (repeatableread)
WHERE   c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

SET     rowcount @cnt

SELECT  @c_id   = c_id,
        @c_balance = c_balance,
        @c_first  = c_first,
        @c_last   = c_last,
        @c_middle = c_middle
FROM    customer WITH (repeatableread)
WHERE   c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

ORDER  BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT  @c_balance = c_balance,
        @c_first   = c_first,
        @c_middle  = c_middle,
        @c_last    = c_last
FROM    customer WITH (repeatableread)
WHERE   c_id = @c_id AND
       c_d_id = @d_id AND
       c_w_id = @w_id

SELECT  @cnt      = @@rowcount
END

-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
RAISERROR('Customer not found',18,1)
GOTO custnotfound
END

-----
-- get order info
-----
SELECT  @o_id      = o_id,
        @o_entry_d = o_entry_d,
        @o_carrier_id = o_carrier_id
FROM    orders WITH (serializable)
WHERE   o_c_id = @c_id AND

```

```

        o_d_id      = @d_id AND
        o_w_id      = @w_id
ORDER BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM   order_line WITH (repeatableread)
WHERE  ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp3tpcc_orderstatus' )
DROP PROCEDURE sp3tpcc_orderstatus
GO

CREATE PROCEDURE sp3tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT @cnt = (count(*)+1)/2
FROM   customer WITH (repeatableread)
WHERE  c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,
       @c_balance = c_balance,
       @c_first = c_first,
       @c_last = c_last,
       @c_middle = c_middle
FROM   customer WITH (repeatableread)
WHERE  c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT @c_balance = c_balance,
       @c_first = c_first,
       @c_middle = c_middle,
       @c_last = c_last
FROM   customer WITH (repeatableread)
WHERE  c_id = @c_id AND
       c_d_id = @d_id AND
       c_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @c_first,

```

```

        @c_middle = @c_middle
FROM   customer WITH (repeatableread)
WHERE  c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT @c_balance = c_balance,
       @c_first = c_first,
       @c_middle = c_middle,
       @c_last = c_last
FROM   customer WITH (repeatableread)
WHERE  c_id = @c_id AND
       c_d_id = @d_id AND
       c_w_id = @w_id

SELECT @cnt = @@rowcount
END

-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
RAISERROR('Customer not found',18,1)
GOTO custnotfound
END

-----
-- get order info
-----
SELECT @o_id = o_id,
       @o_entry_d = o_entry_d,
       @o_carrier_id = o_carrier_id
FROM   orders WITH (serializable)
WHERE  o_c_id = @c_id AND
       o_d_id = @d_id AND
       o_w_id = @w_id
ORDER BY o_id ASC

-----
-- select order lines for the current order
-----
SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM   order_line WITH (repeatableread)
WHERE  ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

-----
-- return data to client
-----
SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,

```

```

GO @o_id

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp4tpcc_orderstatus' )
DROP PROCEDURE sp4tpcc_orderstatus
GO

CREATE PROCEDURE sp4tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT @cnt = (count(*)+1)/2
FROM   customer WITH (repeatableread)
WHERE  c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,
       @c_balance = c_balance,
       @c_first = c_first,
       @c_last = c_last,
       @c_middle = c_middle
FROM   customer WITH (repeatableread)
WHERE  c_last = @c_last AND
       c_w_id = @w_id AND
       c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----
SELECT @c_balance = c_balance,
       @c_first = c_first,
       @c_middle = c_middle,
       @c_last = c_last
FROM   customer WITH (repeatableread)
WHERE  c_id = @c_id AND
       c_d_id = @d_id AND
       c_w_id = @w_id

SELECT @cnt = @@rowcount
END

-----
-- if no such customer
-----
IF (@cnt = 0)
BEGIN
RAISERROR('Customer not found',18,1)

```

```

GOTO custnotfound
END

-----
-- get order info
-----

SELECT @o_id      = o_id,
       @o_entry_d  = o_entry_d,
       @o_carrier_id = o_carrier_id
FROM   orders WITH (serializable)
WHERE  o_c_id      = @c_id AND
       o_d_id      = @d_id AND
       o_w_id      = @w_id
ORDER  BY o_id ASC

-----
-- select order lines for the current order
-----

SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM   order_line WITH (repeatable)
WHERE  ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

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

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp5tpcc_orderstatus' )
DROP PROCEDURE sp5tpcc_orderstatus
GO

CREATE PROCEDURE sp5tpcc_orderstatus
       @w_id int,
       @d_id tinyint,

       @c_id int,
       @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----


```

```

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
       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 TRANSACTION o

```

```

-- return data to client
-----

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp6tpcc_orderstatus' )
DROP PROCEDURE sp6tpcc_orderstatus
GO

CREATE PROCEDURE sp6tpcc_orderstatus
       @w_id int,
       @d_id tinyint,

       @c_id int,
       @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),
        @o_id int,
        @o_entry_d datetime,
        @o_carrier_id smallint,
        @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----

SELECT @cnt = (count(*)+1)/2
FROM customer WITH (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
       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 TRANSACTION o

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

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp7tpcc_orderstatus' )
DROP PROCEDURE sp7tpcc_orderstatus
GO

CREATE PROCEDURE sp7tpcc_orderstatus
    @w_id int,
    @d_id tinyint,

    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
        @c_first char(16),
        @c_middle char(2),

```

```

@o_entry_d intdatetime,
@o_carrier_id smallint,
@cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----

SELECT @cnt = (count(*)+1)/2
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id = @w_id AND
      c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,
       @c_balance = c_balance,
       @c_first = c_first,
       @c_last = c_last,
       @c_middle = c_middle
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id = @w_id AND
      c_d_id = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

SET rowcount 0
END
ELSE
BEGIN
-----
-- get customer info if by id
-----

SELECT @c_balance = c_balance,
       @c_first = c_first,
       @c_middle = c_middle,
       @c_last = c_last
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_d_id = @d_id AND
      c_w_id = @w_id

SELECT @cnt = @@rowcount
END

-----
-- if no such customer
-----

IF (@cnt = 0)
BEGIN
    RAISERROR('Customer not found',18,1)
    GOTO custnotfound
END

-----
-- get order info
-----

SELECT @o_id = o_id,
       @o_entry_d = o_entry_d,
       @o_carrier_id = o_carrier_id
FROM orders WITH (serializable)
WHERE o_c_id = @c_id AND
      o_d_id = @d_id AND
      o_w_id = @w_id
ORDER BY o_id ASC

-----
-- select order lines for the current order
-----

SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,

```

```

ol_delivery_d
FROM order_line WITH (repeatableread)
WHERE ol_o_id = @o_id AND
      ol_d_id = @d_id AND
      ol_w_id = @w_id

custnotfound:

COMMIT TRANSACTION o

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

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,

       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id
GO

```

CreateDeliveryProc.sql

```

-----
--
-- File: DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
--
-- Copyright Microsoft, 2006
-- Creates delivery stored procedure
--
-- Interface Level: 4.20.000
-----

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_delivery' )
DROP PROCEDURE tpcc_delivery
GO

CREATE PROC tpcc_delivery
    @w_id int,
    @o_carrier_id smallint

AS

DECLARE @d_id tinyint,
        @o_id int,
        @c_id int,
        @total money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,

```

```

@oid6 int,
@oid7 int,
@oid8 int,
@oid9 int,
@oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
           @o_id = no_o_id
    FROM new_order WITH (serializable 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 = 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

```

```

@oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
@oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
END

COMMIT TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
           @o_id = no_o_id
    FROM new_order WITH (serializable 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_o_id = @o_id AND

-- 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 TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
        @o_id = no_o_id
    FROM new_order WITH (serializable 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 = c_delivery_cnt + 1
    END
END

```

```

WHERE d_id = @d_id AND
      c_id = @c_id
END

SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
       @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
       @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
       @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
       @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
       @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
       @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
       @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
       @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
       @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
END

COMMIT TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp2tpcc_delivery' )
DROP PROCEDURE sp2tpcc_delivery
GO

CREATE PROC sp2tpcc_delivery
    @w_id int,
    @o_carrier_id smallint
AS

DECLARE @d_id tinyint,
        @o_id int,
        @c_id int,
        @total money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

```

```

SELECT @d_id = 0
BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
        @o_id = no_o_id
    FROM new_order WITH (serializable 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 = c_delivery_cnt + 1
        WHERE c_w_id = @w_id AND
              c_d_id = @d_id AND
              c_id = @c_id
    END

    SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
           @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
           @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
           @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
           @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
           @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
           @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
           @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
           @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
           @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
    END

COMMIT TRANSACTION d

```

```

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
  BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
      @o_id = no_o_id
    FROM new_order WITH (serializable 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 = c_delivery_cnt + 1

        WHERE c_w_id = @w_id AND
              c_d_id = @d_id AND
              c_id = @c_id

        END

        SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
              @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
              @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
              @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
              @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
              @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
              @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
              @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
              @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
              @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
        END

        COMMIT TRANSACTION d

        -- return delivery data to client

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

        SET QUOTED_IDENTIFIER OFF
        GO

        SET ANSI_NULLS ON
        GO

        IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp4tpcc_delivery' )
          DROP PROCEDURE sp4tpcc_delivery
        GO

        CREATE PROC sp4tpcc_delivery

```

```

  SET @c_carrier_id = @o_carrier_id,
  WHERE o_w_id = @w_id AND
        o_d_id = @d_id AND
        o_id = @o_id

  -- set date in all lineitems for this order (and
  sum amounts)
  UPDATE order_line
  SET ol_delivery_d = GETDATE(),
      @total = @total + ol_amount
  WHERE ol_w_id = @w_id AND
        ol_d_id = @d_id AND
        ol_o_id = @o_id

  -- 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 TRANSACTION d

  -- return delivery data to client

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

  SET QUOTED_IDENTIFIER OFF
  GO

  SET ANSI_NULLS ON
  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 smallint
AS
DECLARE @d_id tinyint,
        @o_id int,
        @c_id int,
        @total money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
  BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
      @o_id = no_o_id
    FROM new_order WITH (serializable 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 = c_delivery_cnt + 1

        WHERE c_w_id = @w_id AND
              c_d_id = @d_id AND
              c_id = @c_id

        END

        SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,

```

```

        @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
        @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
        @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
        @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
        @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
        @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
        @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
        @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
        @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
END

COMMIT TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
        SELECT @d_id = @d_id + 1,

```

```

        @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 = c_delivery_cnt + 1

WHERE c_w_id = @w_id AND
        c_d_id = @d_id AND
        c_id = @c_id
END

SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
        @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
        @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
        @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
        @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
        @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
        @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
        @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
        @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
        @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
END

COMMIT TRANSACTION d

-- return delivery data to client

SELECT @oid1,
        @oid2,
        @oid3,

```

```

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
        SELECT @d_id = @d_id + 1,
        @total = 0,
        @o_id = 0

        SELECT TOP 1
        @o_id = no_o_id
FROM new_order WITH (serializable 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 = c_delivery_cnt + 1

WHERE c_w_id = @w_id AND
      c_d_id = @d_id AND
      c_id = @c_id
END

SELECT @oid1 = CASE @d_id WHEN 1 THEN
@o_id ELSE @oid1 END,
       @oid2 = CASE @d_id WHEN 2 THEN
@o_id ELSE @oid2 END,
       @oid3 = CASE @d_id WHEN 3 THEN
@o_id ELSE @oid3 END,
       @oid4 = CASE @d_id WHEN 4 THEN
@o_id ELSE @oid4 END,
       @oid5 = CASE @d_id WHEN 5 THEN
@o_id ELSE @oid5 END,
       @oid6 = CASE @d_id WHEN 6 THEN
@o_id ELSE @oid6 END,
       @oid7 = CASE @d_id WHEN 7 THEN
@o_id ELSE @oid7 END,
       @oid8 = CASE @d_id WHEN 8 THEN
@o_id ELSE @oid8 END,
       @oid9 = CASE @d_id WHEN 9 THEN
@o_id ELSE @oid9 END,
       @oid10 = CASE @d_id WHEN 10 THEN
@o_id ELSE @oid10 END
END

COMMIT TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
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 smallint

```

```

AS
DECLARE @d_id tinyint,
        @o_id int,
        @c_id int,
        @total money,
        @oid1 int,
        @oid2 int,
        @oid3 int,
        @oid4 int,
        @oid5 int,
        @oid6 int,
        @oid7 int,
        @oid8 int,
        @oid9 int,
        @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
WHILE (@d_id < 10)
BEGIN
    SELECT @d_id = @d_id + 1,
           @total = 0,
           @o_id = 0

    SELECT TOP 1
           @o_id = no_o_id
    FROM new_order WITH (serializable 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 = @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 TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_payment' )
DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE tpcc_payment
    @w_id int,

```

CreatePaymentProc.sql

```

-----
--
-- File: PAYMENT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
--
-- Copyright Microsoft, 2006
--
-- Creates payment stored procedure
--
-- Interface Level: 4.20.000
--
-----

```

```

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_payment' )
DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE tpcc_payment
    @w_id int,

```

```

        @c_w_id int,
        @h_amount smallmoney,
        @d_id tinyint,
        @c_d_id tinyint,
        @c_id int,
        @c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,
        @c_credit char(2),
        @c_credit_lim money,
        @c_balance money,
        @c_discount smallmoney,
        @c_data char(42),
        @datetime datetime,
        @w_ytd money,
        @d_ytd money,
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id =
c_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_d_id =
ORDER BY c_last, c_first

```

```

SET rowcount 0
END

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@h_amount,
@c_first = c_first,
@c_middle = c_middle,
@c_last = c_last,
@c_street_1 = c_street_1,
@c_street_2 = c_street_2,
@c_city = c_city,
@c_state = c_state,
@c_zip = c_zip,
@c_phone = c_phone,
@c_credit = c_credit,
@c_credit_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@c_id_local = c_id
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,
@d_zip = d_zip,
@d_name = d_name,
@d_id_local = d_id
WHERE d_w_id = @w_id AND
d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
@w_street_1 = w_street_1,
@w_street_2 = w_street_2,
@w_city = w_city,
@w_state = w_state,
@w_zip = w_zip,
@w_name = w_name,
@w_id_local = w_id
WHERE w_id = @w_id

```

```

-- create history record
INSERT INTO history
VALUES (@c_id_local,
@c_d_id,
@c_w_id,
@d_id_local,
@w_id_local,
@datetime,
@h_amount,
@w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,
@w_city,
@w_state,
@w_zip,
@d_street_1,
@d_street_2,
@d_city,
@d_state,
@d_zip,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,
@c_state,
@c_zip,
@c_phone,
@c_since,
@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp0tpcc_payment' )
DROP PROCEDURE sp0tpcc_payment
GO

CREATE PROCEDURE sp0tpcc_payment
@c_w_id int,
@c_w_id int,
@h_amount smallmoney,
@d_id tinyint,
@c_d_id tinyint,
@c_id int,
@c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),

```

```

@d_zip char(9),
@d_name char(10),
@c_first char(16),
@c_middle char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_since datetime,
@c_credit char(2),
@c_credit_lim money,
@c_balance money,
@c_discount smallmoney,
@c_data char(42),
@datetime datetime,
@w_ytd money,
@d_ytd money,
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id =
@c_w_id AND
c_d_id =
@c_d_id

SELECT @val = (@cnt + 1) / 2

SET rowcount @val

SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id =
c_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_phone = c_phone,
@c_credit = c_credit,
@c_credit_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@c_id_local = c_id
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,
@d_zip = d_zip,
@d_name = d_name,
@d_id_local = d_id
WHERE d_w_id = @w_id AND
d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
@w_street_1 = w_street_1,
@w_street_2 = w_street_2,
@w_city = w_city,
@w_state = w_state,
@w_zip = w_zip,
@w_name = w_name,
@w_id_local = w_id
WHERE w_id = @w_id

-- create history record
INSERT INTO history
VALUES (@c_id_local,
@c_d_id,
@c_w_id,
@d_id_local,
@w_id_local,
@datetime,
@h_amount,
@w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,

```

```

@w_street_2,
@w_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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp1tpcc_payment' )
DROP PROCEDURE sp1tpcc_payment
GO

CREATE PROCEDURE sp1tpcc_payment
@w_id int,
@c_w_id int,
@h_amount smallmoney,
@d_id tinyint,
@c_d_id tinyint,
@c_id int,
@c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
@w_street_2 char(20),
@w_city char(20),
@w_state char(2),
@w_zip char(9),
@w_name char(10),
@d_street_1 char(20),
@d_street_2 char(20),
@d_city char(20),
@d_state char(2),
@d_zip char(9),
@d_name char(10),
@c_first char(16),
@c_middle char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_since datetime,
@c_credit char(2),
@c_credit_lim money,
@c_balance money,
@c_discount smallmoney,
@c_data char(42),
@datetime datetime,
@w_ytd money,
@d_ytd money,
@cnt smallint,

```

```

    @val          smallint,
    @screen_data char(200),
    @d_id_local  tinyint,
    @w_id_local  int,
    @c_id_local  int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id =
@c_w_id AND
      c_d_id =
@c_d_id

SELECT @val = (@cnt + 1) / 2

SET rowcount @val

SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id =
@c_w_id AND
      c_d_id =
ORDER BY c_last, c_first

SET rowcount 0
END

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,
    @c_credit_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @c_id_local = c_id
WHERE c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +

```

```

convert(char(19),@h_amount)
-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
    @screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
    @d_street_1 = d_street_1,
    @d_street_2 = d_street_2,
    @d_city = d_city,
    @d_state = d_state,
    @d_zip = d_zip,
    @d_name = d_name,
    @d_id_local = d_id
WHERE d_w_id = @w_id AND
      d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
    @w_street_1 = w_street_1,
    @w_street_2 = w_street_2,
    @w_city = w_city,
    @w_state = w_state,
    @w_zip = w_zip,
    @w_name = w_name,
    @w_id_local = w_id
WHERE w_id = @w_id

-- create history record
INSERT INTO history
VALUES (@c_id_local,
        @c_d_id,
        @c_w_id,
        @d_id_local,
        @w_id_local,
        @datetime,
        @h_amount,
        @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,

```

```

@c_balance,
@screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp2tpcc_payment' )
DROP PROCEDURE sp2tpcc_payment
GO

CREATE PROCEDURE sp2tpcc_payment
    @w_id int,
    @c_w_id int,
    @h_amount smallmoney,
    @d_id tinyint,
    @c_d_id tinyint,
    @c_id int,
    @c_last char(16) = ""
AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,
        @c_credit char(2),
        @c_credit_lim money,
        @c_balance money,
        @c_discount smallmoney,
        @c_data char(42),
        @datetime datetime,
        @w_ytd money,
        @d_ytd money,
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
      c_w_id =
@c_w_id AND

```

```

c_d_id =
@c_d_id

SELECT @val = (@cnt + 1) / 2

SET rowcount @val

SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND

c_w_id =
@c_w_id AND
c_d_id =
@c_d_id
ORDER BY c_last, c_first

SET rowcount 0
END

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@h_amount,
@c_first = c_first,
@c_middle = c_middle,
@c_last = c_last,
@c_street_1 = c_street_1,
@c_street_2 = c_street_2,
@c_city = c_city,
@c_state = c_state,
@c_zip = c_zip,
@c_phone = c_phone,
@c_credit = c_credit,
@c_credit_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@c_id_local = c_id
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,

```

```

@d_name = d_name,
@d_id_local = d_id
WHERE d_w_id = @w_id AND
d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
@w_street_1 = w_street_1,
@w_street_2 = w_street_2,
@w_city = w_city,
@w_state = w_state,
@w_zip = w_zip,
@w_name = w_name,
@w_id_local = w_id
WHERE w_id = @w_id

-- create history record
INSERT INTO history
VALUES (@c_id_local,
@c_d_id,
@c_w_id,
@d_id_local,
@w_id_local,
@datetime,
@h_amount,
@w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,
@w_city,
@w_state,
@w_zip,
@d_street_1,
@d_street_2,
@d_city,
@d_state,
@d_zip,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,
@c_state,
@c_zip,
@c_phone,
@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp3tpcc_payment' )
DROP PROCEDURE sp3tpcc_payment
GO

CREATE PROCEDURE sp3tpcc_payment
@w_id int,
@c_w_id int,
@h_amount smallmoney,

```

```

@c_d_id tinyint,
@c_id int,
@c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
@w_street_2 char(20),
@w_city char(20),
@w_state char(2),
@w_zip char(9),
@w_name char(10),
@d_street_1 char(20),
@d_street_2 char(20),
@d_city char(20),
@d_state char(2),
@d_zip char(9),
@d_name char(10),
@c_first char(16),
@c_middle char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_since datetime,
@c_credit char(2),
@c_credit_lim money,
@c_balance money,
@c_discount smallmoney,
@c_data char(42),
@datetime datetime,
@w_ytd money,
@d_ytd money,
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND

c_w_id =
@c_w_id AND
c_d_id =
@c_d_id

SELECT @val = (@cnt + 1) / 2

SET rowcount @val

SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND

c_w_id =
@c_w_id AND
c_d_id =
@c_d_id
ORDER BY c_last, c_first

SET rowcount 0
END

```



```

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,
    @c_credit_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @c_id_local = c_id
WHERE c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    -- compute new info
    SELECT @c_data = convert(char(5),@c_id) +
        convert(char(4),@c_d_id) +
        convert(char(5),@c_w_id) +
        convert(char(4),@d_id) +
        convert(char(5),@w_id) +
        convert(char(19),@h_amount)

    -- update customer info
    UPDATE customer
    SET c_data = @c_data +
substring(c_data, 1, 458),
        @screen_data = @c_data +
substring(c_data, 1, 158)
    WHERE c_id = @c_id AND
          c_w_id = @c_w_id AND
          c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
    @d_street_1 = d_street_1,
    @d_street_2 = d_street_2,
    @d_city = d_city,
    @d_state = d_state,
    @d_zip = d_zip,
    @d_name = d_name,
    @d_id_local = d_id
WHERE d_w_id = @w_id AND
      d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
    @w_street_1 = w_street_1,
    @w_street_2 = w_street_2,
    @w_city = w_city,
    @w_state = w_state,
    @w_zip = w_zip,
    @w_name = w_name,
    @w_id_local = w_id
WHERE w_id = @w_id

-- create history record

```

```

INSERT INTO          history
VALUES (@c_id_local, @c_d_id,
        @c_w_id,
        @d_id_local,
        @w_id_local,
        @datetime,
        @h_amount,
        @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_city,
        @w_state,
        @w_zip,
        @d_street_1,
        @d_street_2,
        @d_city,
        @d_state,
        @d_zip,
        @c_first,
        @c_middle,
        @c_street_1,
        @c_street_2,
        @c_city,
        @c_state,
        @c_zip,
        @c_phone,
        @c_since,
        @c_credit,
        @c_credit_lim,
        @c_discount,
        @c_balance,
        @screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
            name = 'sp4tpcc_payment' )
    DROP PROCEDURE sp4tpcc_payment
GO

CREATE PROCEDURE sp4tpcc_payment
    @w_id int,
    @c_w_id int,
    @h_amount smallmoney,
    @d_id tinyint,
    @c_d_id tinyint,
    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),

```

```

        @c_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 money,
        @c_balance money,
        @c_discount smallmoney,
        @c_data char(42),
        @datetime datetime,
        @w_ytd money,
        @d_ytd money,
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
    -- get customer id and info using last name
    SELECT @cnt = COUNT(*)
    FROM customer WITH (repeatable read)
    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 (repeatable read)
    WHERE c_last = @c_last AND
          c_w_id =
          @c_w_id AND
          c_d_id =
          @c_d_id

    SET rowcount 0
END

-- get customer info and update balances

UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
@h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,

```

```

        @c_credit_lim = c_credit_lim,
        @c_discount = c_discount,
        @c_since = c_since,
        @c_id_local = c_id
WHERE   c_id = @c_id AND
        c_w_id = @c_w_id AND
        c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    -- compute new info
    SELECT @c_data = convert(char(5),@c_id) +
        convert(char(4),@c_d_id) +
        convert(char(5),@c_w_id) +
        convert(char(4),@d_id) +
        convert(char(5),@w_id) +
        convert(char(19),@h_amount)

    -- update customer info
    UPDATE customer
    SET   c_data = @c_data +
        substring(c_data, 1, 458),
        @screen_data = @c_data +
        substring(c_data, 1, 158)
    WHERE c_id = @c_id AND
        c_w_id = @c_w_id AND
        c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET   d_ytd = d_ytd + @h_amount,
    @d_street_1 = d_street_1,
    @d_street_2 = d_street_2,
    @d_city = d_city,
    @d_state = d_state,
    @d_zip = d_zip,
    @d_name = d_name,
    @d_id_local = d_id
WHERE d_w_id = @w_id AND
    d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET   w_ytd = w_ytd + @h_amount,
    @w_street_1 = w_street_1,
    @w_street_2 = w_street_2,
    @w_city = w_city,
    @w_state = w_state,
    @w_zip = w_zip,
    @w_name = w_name,
    @w_id_local = w_id
WHERE w_id = @w_id

-- create history record
INSERT INTO          history
VALUES (@c_id_local,
        @c_d_id,
        @c_w_id,
        @d_id_local,
        @w_id_local,
        @datetime,
        @h_amount,
        @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
    @c_last,
    @datetime,
    @w_street_1,
    @w_street_2,
    @w_city,

```

```

    @w_state,
    @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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp5tpcc_payment' )
DROP PROCEDURE sp5tpcc_payment
GO

CREATE PROCEDURE sp5tpcc_payment
    @w_id int,
    @c_w_id int,
    @h_amount smallmoney,
    @d_id tinyint,
    @c_d_id tinyint,
    @c_id int,
    @c_last char(16) = ""
AS
DECLARE @w_street_1 char(20),
    @w_street_2 char(20),
    @w_city char(20),
    @w_state char(2),
    @w_zip char(9),
    @w_name char(10),
    @d_street_1 char(20),
    @d_street_2 char(20),
    @d_city char(20),
    @d_state char(2),
    @d_zip char(9),
    @d_name char(10),
    @c_first char(16),
    @c_middle char(2),
    @c_street_1 char(20),
    @c_street_2 char(20),
    @c_city char(20),
    @c_state char(2),
    @c_zip char(9),
    @c_phone char(16),
    @c_since datetime,
    @c_credit char(2),
    @c_credit_lim money,
    @c_balance money,
    @c_discount smallmoney,
    @c_data char(42),
    @datetime datetime,
    @w_ytd money,
    @d_ytd money,
    @cnt smallint,
    @val smallint,

```

```

    @c_id_local tinyint(200),
    @w_id_local int,
    @c_id_local int
SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
    -- get customer id and info using last name
    SELECT @cnt = COUNT(*)
    FROM customer WITH (repeatableread)
    WHERE c_last = @c_last AND
        c_w_id =
        @c_w_id AND
        c_d_id =
        @c_d_id

    SELECT @val = (@cnt + 1) / 2

    SET rowcount @val

    SELECT @c_id = c_id
    FROM customer WITH (repeatableread)
    WHERE c_last = @c_last AND
        c_w_id =
        @c_w_id AND
        c_d_id =
        @c_d_id
    ORDER BY c_last, c_first

    SET rowcount 0
END

-- get customer info and update balances

UPDATE customer
SET   @c_balance = c_balance = c_balance -
    @h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment +
    @h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,
    @c_credit_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @c_id_local = c_id
WHERE c_id = @c_id AND
    c_w_id = @c_w_id AND
    c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    -- compute new info
    SELECT @c_data = convert(char(5),@c_id) +
        convert(char(4),@c_d_id) +
        convert(char(5),@c_w_id) +
        convert(char(4),@d_id) +
        convert(char(5),@w_id) +
        convert(char(19),@h_amount)

```

```

-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,
@d_zip = d_zip,
@d_name = d_name,
@d_id_local = d_id
WHERE d_w_id = @w_id AND
d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET w_ytd = w_ytd + @h_amount,
@w_street_1 = w_street_1,
@w_street_2 = w_street_2,
@w_city = w_city,
@w_state = w_state,
@w_zip = w_zip,
@w_name = w_name,
@w_id_local = w_id
WHERE w_id = @w_id

-- create history record
INSERT INTO history
VALUES (@c_id_local,
@c_d_id,
@c_w_id,
@d_id_local,
@w_id_local,
@datetime,
@h_amount,
@w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,
@w_city,
@w_state,
@w_zip,
@d_street_1,
@d_street_2,
@d_city,
@d_state,
@d_zip,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,
@c_state,
@c_zip,
@c_phone,
@c_since,
@c_credit,
@c_credit_lim,
@c_discount,

```

```

@c_data
GO
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS ON
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp6tpcc_payment' )
DROP PROCEDURE sp6tpcc_payment
GO
CREATE PROCEDURE sp6tpcc_payment
@w_id int,
@c_w_id int,
@h_amount smallmoney,
@d_id tinyint,
@c_d_id tinyint,
@c_id int,
@c_last char(16) = ""
AS
DECLARE @w_street_1 char(20),
@w_street_2 char(20),
@w_city char(20),
@w_state char(2),
@w_zip char(9),
@w_name char(10),
@d_street_1 char(20),
@d_street_2 char(20),
@d_city char(20),
@d_state char(2),
@d_zip char(9),
@d_name char(10),
@c_first char(16),
@c_middle char(2),
@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_since datetime,
@c_credit char(2),
@c_credit_lim money,
@c_balance money,
@c_discount smallmoney,
@c_data char(42),
@datetime datetime,
@w_ytd money,
@d_ytd money,
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int
SELECT @screen_data = ""
BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()
IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id =
@c_w_id AND

```

```

c_d_id =
@c_d_id
SELECT @val = (@cnt + 1) / 2
SET rowcount @val
SELECT @c_id = c_id
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
c_w_id =
@c_w_id AND
c_d_id =
@c_d_id
ORDER BY c_last, c_first
SET rowcount 0
END
-- get customer info and update balances
UPDATE customer
SET @c_balance = c_balance = c_balance -
@h_amount,
c_payment_cnt = c_payment_cnt + 1,
c_ytd_payment = c_ytd_payment +
@h_amount,
@c_first = c_first,
@c_middle = c_middle,
@c_last = c_last,
@c_street_1 = c_street_1,
@c_street_2 = c_street_2,
@c_city = c_city,
@c_state = c_state,
@c_zip = c_zip,
@c_phone = c_phone,
@c_credit = c_credit,
@c_credit_lim = c_credit_lim,
@c_discount = c_discount,
@c_since = c_since,
@c_id_local = c_id
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)
-- update customer info
UPDATE customer
SET c_data = @c_data +
substring(c_data, 1, 458),
@screen_data = @c_data +
substring(c_data, 1, 158)
WHERE c_id = @c_id AND
c_w_id = @c_w_id AND
c_d_id = @c_d_id
END
-- get district data and update year-to-date
UPDATE district
SET d_ytd = d_ytd + @h_amount,
@d_street_1 = d_street_1,
@d_street_2 = d_street_2,
@d_city = d_city,
@d_state = d_state,
@d_zip = d_zip,

```

```

        @d_name = d_name,
        @d_id_local = d_id
WHERE   d_w_id = @w_id AND
        d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET   w_ytd = w_ytd + @h_amount,
      @w_street_1 = w_street_1,
      @w_street_2 = w_street_2,
      @w_city = w_city,
      @w_state = w_state,
      @w_zip = w_zip,
      @w_name = w_name,
      @w_id_local = w_id
WHERE  w_id = @w_id

-- create history record
INSERT INTO          history
VALUES (@c_id_local,
        @c_d_id,
        @c_w_id,
        @d_id_local,
        @w_id_local,
        @datetime,
        @h_amount,
        @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,
       @c_discount,
       @c_balance,
       @screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp7tpcc_payment' )
DROP PROCEDURE sp7tpcc_payment
GO

CREATE PROCEDURE sp7tpcc_payment
        @w_id int,
        @c_w_id int,
        @h_amount smallmoney,

```

```

        @c_d_id tinyint,
        @c_id int,
        @c_last char(16) = ""

AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,
        @c_credit char(2),
        @c_credit_lim money,
        @c_balance money,
        @c_discount smallmoney,
        @c_data char(42),
        @datetime datetime,
        @w_ytd money,
        @d_ytd money,
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

SELECT @screen_data = ""

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

IF (@c_id = 0)
BEGIN
-- get customer id and info using last name
SELECT @cnt = COUNT(*)
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
@w_id = @c_w_id AND
@c_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,
      @c_id_local = c_id
WHERE  c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
-- compute new info
SELECT @c_data = convert(char(5),@c_id) +
convert(char(4),@c_d_id) +
convert(char(5),@c_w_id) +
convert(char(4),@d_id) +
convert(char(5),@w_id) +
convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET   c_data = @c_data +
substring(c_data, 1, 458),
      @screen_data = @c_data +
substring(c_data, 1, 158)
WHERE  c_id = @c_id AND
      c_w_id = @c_w_id AND
      c_d_id = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET   d_ytd = d_ytd + @h_amount,
      @d_street_1 = d_street_1,
      @d_street_2 = d_street_2,
      @d_city = d_city,
      @d_state = d_state,
      @d_zip = d_zip,
      @d_name = d_name,
      @d_id_local = d_id
WHERE  d_w_id = @w_id AND
      d_id = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET   w_ytd = w_ytd + @h_amount,
      @w_street_1 = w_street_1,
      @w_street_2 = w_street_2,
      @w_city = w_city,
      @w_state = w_state,
      @w_zip = w_zip,
      @w_name = w_name,
      @w_id_local = w_id
WHERE  w_id = @w_id

-- create history record

```

```

INSERT INTO          history
VALUES (@c_id_local,
        @c_d_id,
        @c_w_id,
        @d_id_local,
        @w_id_local,
        @datetime,
        @h_amount,
        @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,
       @c_discount,
       @c_balance,
       @screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

CreateStocklevProc.sql

```

-----
--
-- File: STOCKLEV.SQL
--       Microsoft TPC-C Benchmark Kit Ver. 4.68
--
-- Copyright Microsoft, 2006
--
--       Creates stock level stored procedure
--
-- Interface Level: 4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

```

```

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'tpcc_stocklevel' )
DROP PROCEDURE tpcc_stocklevel
GO

CREATE PROCEDURE tpcc_stocklevel
        @w_id int,
        @d_id tinyint,
        @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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp0tpcc_stocklevel' )
DROP PROCEDURE sp0tpcc_stocklevel
GO

CREATE PROCEDURE 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),
        @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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp1tpcc_stocklevel' )
DROP PROCEDURE sp1tpcc_stocklevel
GO

```

```

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp2tpcc_stocklevel' )
DROP PROCEDURE sp2tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp3tpcc_stocklevel' )
DROP PROCEDURE sp3tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp4tpcc_stocklevel' )
DROP PROCEDURE sp4tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp5tpcc_stocklevel' )
DROP PROCEDURE sp5tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp6tpcc_stocklevel' )
DROP PROCEDURE sp6tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE
name = 'sp7tpcc_stocklevel' )
DROP PROCEDURE sp7tpcc_stocklevel
GO

CREATE PROCEDURE 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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

Appendix C Tunable Parameters

Disabled Windows Services

Application Experience Lookup Service
 Base Filtering Engine
 Computer Browser
 Distributed Link Tracking Server
 Emulex HBA Management
 Emulex Service Manager
 HP SmartArray SAS/SATA Event Notification
 Internet Connection Service
 Error Reporting Service
 IPsec Policy Agent
 PnP-X IP Bus Enumerator
 Print Spooler
 Remote Registry
 Routing and Remote Access
 Shell Detection Service
 SQLServer Active Directory Helper
 SQL Server Browser
 SQL Server Full Text Search
 SQL Server VSS Writer
 SSDP Discovery
 UPnP Device Host
 Windows Firewall/Internet Connection Sharing
 Windows Update
 WinHTTP Web Proxy Auto-Discovery Service

Server System Configuration

System Information report written at: 01/13/11 15:37:16
 System Name: SQLWESTMERE
 [System Summary]

Item	Value
OS Name	Microsoft Windows Server 2008 R2 Enterprise
Version	6.1.7600 Build 7600
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	SQLWESTMERE
System Manufacturer	HP
System Model	ProLiant BL685c G7

System Type x64-based PC
 Processor AMD Opteron(tm) Processor 6174, 2200 Mhz, 12 Core(s), 12 Logical Processor(s)
 Processor AMD Opteron(tm) Processor 6174, 2200 Mhz, 12 Core(s), 12 Logical Processor(s)
 Processor AMD Opteron(tm) Processor 6174, 2200 Mhz, 12 Core(s), 12 Logical Processor(s)

Processor AMD Opteron(tm) Processor 6174, 2200 Mhz, 12 Core(s), 12 Logical Processor(s)
 BIOS Version/Date HP A20, 9/30/2010

SMBIOS Version 2.7
 Windows Directory C:\Windows
 System Directory C:\Windows\system32

Boot Device \Device\HarddiskVolume18
 Locale United States
 Hardware Abstraction Layer Version = "6.1.7600.16385"
 User Name Not Available
 Time Zone Pacific Standard Time
 Installed Physical Memory (RAM) 512 GB

Total Physical Memory 512 GB
 Available Physical Memory 14.7 GB
 Total Virtual Memory 514 GB
 Available Virtual Memory 13.5 GB
 Page File Space 2.00 GB
 Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
 Memory Address 0xFD800000-0xFD8FFFFFF
 QLogic Fibre Channel Adapter

Memory Address 0xFD800000-0xFD8FFFFFF
 PCI standard PCI-to-PCI bridge

IRQ 52 PCI standard PCI-to-PCI bridge

IRQ 52 PCI standard PCI-to-PCI bridge

Memory Address 0xF0000000-0xF7FFFFFF
 ATI I/O Communications Processor
 PCI Bus Controller
 Memory Address 0xF0000000-0xF7FFFFFF
 Standard VGA Graphics Adapter

I/O Port 0x00000000-0x0000000F Direct
 memory access controller
 I/O Port 0x00000000-0x0000000F PCI bus

I/O Port 0x000003C0-0x000003DF PCI bus

I/O Port 0x000003C0-0x000003DF Standard
 VGA Graphics Adapter

Memory Address 0xFDC00000-0xFDCFFFFFF
 QLogic Fibre Channel Adapter

Memory Address 0xFDC00000-0xFDCFFFFFF
 PCI standard PCI-to-PCI bridge

IRQ 86 Microsoft ACPI-Compliant System

IRQ 86 PCI standard PCI-to-PCI bridge

I/O Port 0x00000070-0x00000071 System
 CMOS/real time clock
 I/O Port 0x00000070-0x00000071
 Motherboard resources

I/O Port 0x00002000-0x00002FFF ATI I/O
 Communications Processor PCI Bus Controller

I/O Port 0x00002000-0x00002FFF Standard
 VGA Graphics Adapter

IRQ 10 Mass Storage Controller
 IRQ 10 Mass Storage Controller

IRQ 54 PCI standard PCI-to-PCI bridge

IRQ 54 PCI standard PCI-to-PCI bridge

IRQ 11 Mass Storage Controller
 IRQ 11 Mass Storage Controller

IRQ 22 Standard OpenHCD USB Host
 Controller

IRQ 22 Standard Enhanced PCI to USB
 Host Controller

IRQ 22 Standard OpenHCD USB Host
 Controller

IRQ 44 HP ProLiant iLO 3 Management
 Controller (CHIF)

IRQ 44 Standard Universal PCI to USB Host
 Controller

IRQ 23 Standard OpenHCD USB Host
 Controller

IRQ 23 Standard OpenHCD USB Host
 Controller

IRQ 23 Standard Enhanced PCI to USB
 Host Controller

I/O Port 0x00006000-0x0000FFFF PCI bus

I/O Port 0x00006000-0x0000FFFF PCI
 standard PCI-to-PCI bridge

I/O Port 0x00003000-0x00003FFF PCI
 standard PCI-to-PCI bridge

I/O Port 0x00003000-0x00003FFF HP
 ProLiant iLO 3 Management Controller (CORE)

Memory Address 0xFED00000-0xFED003FF
 High precision event timer

Memory Address 0xFED00000-0xFED003FF
 PCI bus

Memory Address 0xFED00000-0xFED003FF
 PCI bus

Memory Address 0xFD300000-0xFD6FFFFFF
 PCI standard PCI-to-PCI bridge

Memory Address 0xFD300000-0xFD6FFFFFF
 QLogic Fibre Channel Adapter

I/O Port 0x000000A0-0x000000BF
 Motherboard resources

I/O Port 0x000000A0-0x000000BF
 Programmable interrupt controller

Memory Address 0xA0000-0xBFFFF PCI bus

Memory Address 0xA0000-0xBFFFF Standard
 VGA Graphics Adapter

I/O Port 0x000003B0-0x000003BB PCI bus

I/O Port 0x00003B0-0x00003BB	Standard	VGA Graphics Adapter
I/O Port 0x00001000-0x00001007	Standard	Dual Channel PCI IDE Controller
I/O Port 0x00001000-0x00001007	PCI bus	
I/O Port 0x0000600-0x000061F		Extended IO Bus
I/O Port 0x0000600-0x000061F		Motherboard resources
I/O Port 0x00000020-0x0000003F		Motherboard resources
I/O Port 0x00000020-0x0000003F		Programmable interrupt controller
IRQ 84		Microsoft ACPI-Compliant System
IRQ 84		PCI standard PCI-to-PCI bridge
[DMA]		
Resource Channel 7	Device	Status
	Direct memory access controller	OK
[Forced Hardware]		
Device	PNP Device ID	
[I/O]		
Resource 0x00004000-0x00004FFF	Device	Status
	PCI standard PCI-to-PCI bridge	OK
0x00000040-0x00000043		System timer OK
0x00000000-0x0000000F		Direct memory access controller OK
0x00000000-0x0000000F		PCI bus OK
0x00000080-0x0000008F		Direct memory access controller OK
0x000000C0-0x000000DF		Direct memory access controller OK
0x00003800-0x000038FF		HP ProLiant iLO 3 Management Controller (CHIF) OK
0x00007000-0x00007FFF		PCI standard PCI-to-PCI bridge OK
0x00000060-0x00000060		Standard PS/2 Keyboard OK
0x00000064-0x00000064		Standard PS/2 Keyboard OK
0x00002000-0x00002FFF		ATI I/O Communications Processor PCI Bus Controller OK
0x00002000-0x00002FFF		Standard VGA Graphics Adapter OK
0x000002F8-0x000002FF		Communications Port (COM2) OK
0x00000061-0x00000061		System speaker OK
0x00000500-0x0000050F		Standard Dual Channel PCI IDE Controller OK
0x0000002E-0x0000002F		Extended IO Bus OK
0x00000620-0x0000065F		Extended IO Bus OK

0x0000680-0x000069F		Extended IO Bus
0x0000600-0x000061F		Extended IO Bus OK
0x0000600-0x000061F		Motherboard resources OK
0x0000660-0x000067F		Extended IO Bus OK
0x0000300-0x000031F		Extended IO Bus OK
0x00001000-0x00001007		Standard Dual Channel PCI IDE Controller OK
0x00001000-0x00001007		PCI bus OK
0x00001008-0x0000100B		Standard Dual Channel PCI IDE Controller OK
0x00001010-0x00001017		Standard Dual Channel PCI IDE Controller OK
0x00001018-0x0000101B		Standard Dual Channel PCI IDE Controller OK
0x00001020-0x0000102F		Standard Dual Channel PCI IDE Controller OK
0x00003000-0x00003FFF		PCI standard PCI-to-PCI bridge OK
0x00003000-0x00003FFF		HP ProLiant iLO 3 Management Controller (CORE) OK
0x000003E0-0x00000CF7		PCI bus OK
0x00000D00-0x00000FFF		PCI bus OK
0x000003B0-0x000003BB		PCI bus OK
0x000003B0-0x000003BB		Standard VGA Graphics Adapter OK
0x000003C0-0x000003DF		PCI bus OK
0x000003C0-0x000003DF		Standard VGA Graphics Adapter OK
0x00003C00-0x00003C1F		Standard Universal PCI to USB Host Controller OK
0x00006000-0x00006FFF		PCI bus OK
0x00006000-0x00006FFF		PCI standard PCI-to-PCI bridge OK
0x00005000-0x00005FFF		PCI standard PCI-to-PCI bridge OK
0x000001F0-0x000001F7		ATA Channel 0 OK
0x000003F6-0x000003F6		ATA Channel 0 OK
0x00000070-0x00000071		System CMOS/real time clock OK
0x00000070-0x00000071		Motherboard resources OK
0x00000170-0x00000177		ATA Channel 1 OK
0x00000376-0x00000376		ATA Channel 1 OK
0x00000010-0x0000001F		Motherboard resources OK
0x00000020-0x0000003F		Motherboard resources OK
0x00000020-0x0000003F		Programmable interrupt controller OK
0x000000A0-0x000000BF		Motherboard resources OK
0x000000A0-0x000000BF		Programmable interrupt controller OK
0x00000050-0x00000053		Motherboard resources OK
0x00000090-0x0000009F		Motherboard resources OK
0x000000F0-0x000000F0		Motherboard resources OK

0x00000379-0x0000037A		Motherboard resources OK
0x00000400-0x0000043F		Motherboard resources OK
0x000004D0-0x000004D1		Motherboard resources OK
0x000004D6-0x000004D6		Motherboard resources OK
0x00000520-0x00000520		Motherboard resources OK
0x00000580-0x0000059F		Motherboard resources OK
0x00000700-0x00000703		Motherboard resources OK
0x00000820-0x0000082F		Motherboard resources OK
0x00000900-0x000009FE		Motherboard resources OK
0x00000C06-0x00000C07		Motherboard resources OK
0x00000C14-0x00000C14		Motherboard resources OK
0x00000C4A-0x00000C4A		Motherboard resources OK
0x00000C50-0x00000C52		Motherboard resources OK
0x00000C6C-0x00000C6C		Motherboard resources OK
0x00000C6F-0x00000C6F		Motherboard resources OK
0x00000C80-0x00000C83		Motherboard resources OK
0x00000C90-0x00000C9F		Motherboard resources OK
0x00000CA0-0x00000CA5		Motherboard resources OK
0x00000CD0-0x00000CDF		Motherboard resources OK
0x00000F50-0x00000F58		Motherboard resources OK
0x00000B00-0x00000B3F		Motherboard resources OK
0x000003F8-0x000003FF		Motherboard resources OK
0x00000CA2-0x00000CA3		Microsoft Generic IPMI Compliant Device OK
0x00000C00-0x00000C01		Programmable interrupt controller OK
0x00003400-0x000034FF		HP ProLiant iLO 3 Management Controller (CORE) OK
[IRQs]		
Resource IRQ 54	Device	Status
	PCI standard PCI-to-PCI bridge	OK
IRQ 54		PCI standard PCI-to-PCI bridge OK
IRQ 0		System timer OK
IRQ 81		Microsoft ACPI-Compliant System OK
IRQ 82		Microsoft ACPI-Compliant System OK
IRQ 83		Microsoft ACPI-Compliant System OK
IRQ 84		Microsoft ACPI-Compliant System OK
IRQ 84		PCI standard PCI-to-PCI bridge OK
IRQ 85		Microsoft ACPI-Compliant System OK
IRQ 86		Microsoft ACPI-Compliant System OK
IRQ 86		PCI standard PCI-to-PCI bridge OK

IRQ 4294967213 QLogic Fibre Channel Adapter OK
 IRQ 4294967212 QLogic Fibre Channel Adapter OK
 IRQ 4294967211 QLogic Fibre Channel Adapter OK
 IRQ 4294967210 QLogic Fibre Channel Adapter OK
 IRQ 4294967209 QLogic Fibre Channel Adapter OK
 IRQ 4294967208 QLogic Fibre Channel Adapter OK
 IRQ 4294967207 QLogic Fibre Channel Adapter OK
 IRQ 4294967206 QLogic Fibre Channel Adapter OK
 IRQ 4294967205 QLogic Fibre Channel Adapter OK
 IRQ 4294967204 QLogic Fibre Channel Adapter OK
 IRQ 4294967203 QLogic Fibre Channel Adapter OK
 IRQ 4294967202 QLogic Fibre Channel Adapter OK
 IRQ 4294967201 QLogic Fibre Channel Adapter OK
 IRQ 4294967200 QLogic Fibre Channel Adapter OK
 IRQ 4294967199 QLogic Fibre Channel Adapter OK
 IRQ 4294967198 QLogic Fibre Channel Adapter OK
 IRQ 4294967197 QLogic Fibre Channel Adapter OK
 IRQ 4294967196 QLogic Fibre Channel Adapter OK
 IRQ 4294967195 QLogic Fibre Channel Adapter OK
 IRQ 4294967194 QLogic Fibre Channel Adapter OK
 IRQ 4294967193 QLogic Fibre Channel Adapter OK
 IRQ 4294967192 QLogic Fibre Channel Adapter OK
 IRQ 4294967191 QLogic Fibre Channel Adapter OK
 IRQ 4294967070 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967069 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967068 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967067 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967066 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967065 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967064 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967063 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967062 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK

IRQ 4294967061 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967060 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 4294967059 HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 IRQ 23 Standard OpenHCD USB Host Controller OK
 IRQ 23 Standard OpenHCD USB Host Controller OK
 IRQ 23 Standard Enhanced PCI to USB Host Controller OK
 IRQ 4294967294 Smart Array P410i Controller OK
 IRQ 4294967293 Smart Array P410i Controller OK
 IRQ 4294967292 Smart Array P410i Controller OK
 IRQ 4294967291 Smart Array P410i Controller OK
 IRQ 4294967290 Smart Array P410i Controller OK
 IRQ 4294967289 Smart Array P410i Controller OK
 IRQ 4294967288 Smart Array P410i Controller OK
 IRQ 4294967287 Smart Array P410i Controller OK
 IRQ 44 HP ProLiant iLO 3 Management Controller (CHIF) OK
 IRQ 44 Standard Universal PCI to USB Host Controller OK
 IRQ 4294967190 QLogic Fibre Channel Adapter OK
 IRQ 4294967189 QLogic Fibre Channel Adapter OK
 IRQ 4294967188 QLogic Fibre Channel Adapter OK
 IRQ 4294967187 QLogic Fibre Channel Adapter OK
 IRQ 4294967186 QLogic Fibre Channel Adapter OK
 IRQ 4294967185 QLogic Fibre Channel Adapter OK
 IRQ 4294967184 QLogic Fibre Channel Adapter OK
 IRQ 4294967183 QLogic Fibre Channel Adapter OK
 IRQ 4294967182 QLogic Fibre Channel Adapter OK
 IRQ 4294967181 QLogic Fibre Channel Adapter OK
 IRQ 4294967180 QLogic Fibre Channel Adapter OK
 IRQ 4294967179 QLogic Fibre Channel Adapter OK
 IRQ 4294967178 QLogic Fibre Channel Adapter OK
 IRQ 4294967177 QLogic Fibre Channel Adapter OK
 IRQ 4294967176 QLogic Fibre Channel Adapter OK
 IRQ 4294967175 QLogic Fibre Channel Adapter OK
 IRQ 4294967174 QLogic Fibre Channel Adapter OK
 IRQ 4294967173 QLogic Fibre Channel Adapter OK
 IRQ 4294967172 QLogic Fibre Channel Adapter OK
 IRQ 4294967171 QLogic Fibre Channel Adapter OK

IRQ 4294967170 QLogic Fibre Channel Adapter OK
 IRQ 4294967169 QLogic Fibre Channel Adapter OK
 IRQ 4294967168 QLogic Fibre Channel Adapter OK
 IRQ 4294967167 QLogic Fibre Channel Adapter OK
 IRQ 4294967166 QLogic Fibre Channel Adapter OK
 IRQ 4294967165 QLogic Fibre Channel Adapter OK
 IRQ 4294967164 QLogic Fibre Channel Adapter OK
 IRQ 4294967163 QLogic Fibre Channel Adapter OK
 IRQ 4294967162 QLogic Fibre Channel Adapter OK
 IRQ 4294967161 QLogic Fibre Channel Adapter OK
 IRQ 4294967160 QLogic Fibre Channel Adapter OK
 IRQ 4294967159 QLogic Fibre Channel Adapter OK
 IRQ 10 Mass Storage Controller OK
 IRQ 10 Mass Storage Controller OK
 IRQ 22 Standard OpenHCD USB Host Controller OK
 IRQ 22 Standard Enhanced PCI to USB Host Controller OK
 IRQ 1 Standard PS/2 Keyboard OK
 IRQ 3 Communications Port (COM2) OK
 IRQ 52 PCI standard PCI-to-PCI bridge OK
 IRQ 52 PCI standard PCI-to-PCI bridge OK
 IRQ 11 Mass Storage Controller OK
 IRQ 11 Mass Storage Controller OK
 IRQ 4294967158 QLogic Fibre Channel Adapter OK
 IRQ 4294967157 QLogic Fibre Channel Adapter OK
 IRQ 4294967156 QLogic Fibre Channel Adapter OK
 IRQ 4294967155 QLogic Fibre Channel Adapter OK
 IRQ 4294967154 QLogic Fibre Channel Adapter OK
 IRQ 4294967153 QLogic Fibre Channel Adapter OK
 IRQ 4294967152 QLogic Fibre Channel Adapter OK
 IRQ 4294967151 QLogic Fibre Channel Adapter OK
 IRQ 4294967150 QLogic Fibre Channel Adapter OK
 IRQ 4294967149 QLogic Fibre Channel Adapter OK
 IRQ 4294967148 QLogic Fibre Channel Adapter OK
 IRQ 4294967147 QLogic Fibre Channel Adapter OK
 IRQ 4294967146 QLogic Fibre Channel Adapter OK
 IRQ 4294967145 QLogic Fibre Channel Adapter OK
 IRQ 4294967144 QLogic Fibre Channel Adapter OK

IRQ 4294967087 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #4 OK
 IRQ 4294967086 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #4 OK
 IRQ 4294967085 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #4 OK
 IRQ 4294967084 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #4 OK
 IRQ 4294967083 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #4 OK
 IRQ 15 ATA Channel 1 OK
 IRQ 47 HP ProLiant iLO 3 Management Controller (CORE) OK
 IRQ 4294967254 QLogic Fibre Channel Adapter OK
 IRQ 4294967253 QLogic Fibre Channel Adapter OK
 IRQ 4294967252 QLogic Fibre Channel Adapter OK
 IRQ 4294967251 QLogic Fibre Channel Adapter OK
 IRQ 4294967250 QLogic Fibre Channel Adapter OK
 IRQ 4294967249 QLogic Fibre Channel Adapter OK
 IRQ 4294967248 QLogic Fibre Channel Adapter OK
 IRQ 4294967247 QLogic Fibre Channel Adapter OK
 IRQ 4294967246 QLogic Fibre Channel Adapter OK
 IRQ 4294967245 QLogic Fibre Channel Adapter OK
 IRQ 4294967244 QLogic Fibre Channel Adapter OK
 IRQ 4294967243 QLogic Fibre Channel Adapter OK
 IRQ 4294967242 QLogic Fibre Channel Adapter OK
 IRQ 4294967241 QLogic Fibre Channel Adapter OK
 IRQ 4294967240 QLogic Fibre Channel Adapter OK
 IRQ 4294967239 QLogic Fibre Channel Adapter OK
 IRQ 4294967238 QLogic Fibre Channel Adapter OK
 IRQ 4294967237 QLogic Fibre Channel Adapter OK
 IRQ 4294967236 QLogic Fibre Channel Adapter OK
 IRQ 4294967235 QLogic Fibre Channel Adapter OK
 IRQ 4294967234 QLogic Fibre Channel Adapter OK
 IRQ 4294967233 QLogic Fibre Channel Adapter OK
 IRQ 4294967232 QLogic Fibre Channel Adapter OK
 IRQ 4294967231 QLogic Fibre Channel Adapter OK
 IRQ 4294967230 QLogic Fibre Channel Adapter OK
 IRQ 4294967229 QLogic Fibre Channel Adapter OK
 IRQ 4294967228 QLogic Fibre Channel Adapter OK
 IRQ 4294967227 QLogic Fibre Channel Adapter OK

IRQ 4294967226 QLogic Fibre Channel Adapter OK
 IRQ 4294967225 QLogic Fibre Channel Adapter OK
 IRQ 4294967224 QLogic Fibre Channel Adapter OK
 IRQ 4294967223 QLogic Fibre Channel Adapter OK
 IRQ 12 PS/2 Compatible Mouse OK
 IRQ 4294967058 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967057 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967056 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967055 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967054 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967053 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967052 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967051 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967050 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967049 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967048 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 IRQ 4294967047 HP NC551i Dual Port FlexFabric 10Gb Network Adapter #2 OK
 [Memory]
 Resource Device Status
 0xFC700000-0xFCBFFFFF PCI standard PCI-to-PCI bridge OK
 0xFED00000-0xFED003FF High precision event timer OK
 0xFED00000-0xFED003FF PCI bus OK
 0xFED00000-0xFED003FF PCI bus OK
 0xFDF00000-0xFDF3FFFF QLogic Fibre Channel Adapter OK
 0xFDE00000-0xFDEFFFFF QLogic Fibre Channel Adapter OK
 0xFCF90000-0xFCF93FFF HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 0xFCF60000-0xFCF7FFFF HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 0xFCF40000-0xFCF5FFFF HP NC551i Dual Port FlexFabric 10Gb Network Adapter OK
 0xFC2B0000-0xFC2B0FFF Standard OpenHCD USB Host Controller OK
 0xFC800000-0xFCBFFFFF Smart Array P410i Controller OK

0xFC7F0000-0xFC7F0FFF Smart Array P410i Controller OK
 0xFC6E0000-0xFC6E00FF HP ProLiant iLO 3 Management Controller (CHIF) OK
 0xFC500000-0xFC5FFFFF HP ProLiant iLO 3 Management Controller (CHIF) OK
 0xFC480000-0xFC4FFFFF HP ProLiant iLO 3 Management Controller (CHIF) OK
 0xFC470000-0xFC477FFF HP ProLiant iLO 3 Management Controller (CHIF) OK
 0xFC460000-0xFC467FFF HP ProLiant iLO 3 Management Controller (CHIF) OK
 0xFDDF0000-0xFDDF3FFF QLogic Fibre Channel Adapter OK
 0xFDC00000-0xFDCFFFFF QLogic Fibre Channel Adapter OK
 0xFDC00000-0xFDCFFFFF PCI standard PCI-to-PCI bridge OK
 0xFD230000-0xFD233FFF Mass Storage Controller OK
 0xFD200000-0xFD21FFFF Mass Storage Controller OK
 0xFD1E0000-0xFD1FFFFF Mass Storage Controller OK
 0xFC2D0000-0xFC2D0FFF Standard OpenHCD USB Host Controller OK
 0xFD1D0000-0xFD1D3FFF Mass Storage Controller OK
 0xFD1A0000-0xFD1BFFFF Mass Storage Controller OK
 0xFD180000-0xFD19FFFF Mass Storage Controller OK
 0xFC300000-0xFC3FFFFF ATI I/O Communications Processor PCI Bus Controller OK
 0xF0000000-0xF7FFFFFFF ATI I/O Communications Processor PCI Bus Controller OK
 0xF0000000-0xF7FFFFFFF Standard VGA Graphics Adapter OK
 0xFC2A0000-0xFC2A0FFF Standard OpenHCD USB Host Controller OK
 0xFCD00000-0xFCFFFFFFF PCI standard PCI-to-PCI bridge OK
 0xFCF30000-0xFCF33FFF Mass Storage Controller OK
 0xFCF00000-0xFCF1FFFF Mass Storage Controller OK
 0xFCEE0000-0xFCEFFFFFFF Mass Storage Controller OK
 0xFD000000-0xFD2FFFFF PCI standard PCI-to-PCI bridge OK
 0xFDBF0000-0xFDBF3FFF QLogic Fibre Channel Adapter OK
 0xFDA00000-0xFDAFFFFFFF QLogic Fibre Channel Adapter OK
 0xFC2F0000-0xFC2F03FF Standard Dual Channel PCI IDE Controller OK
 0xFCED0000-0xFCED3FFF Mass Storage Controller OK
 0xFCEA0000-0xFCEBFFFF Mass Storage Controller OK
 0xFCE80000-0xFCE9FFFF Mass Storage Controller OK
 0xFC400000-0xFC6FFFFF PCI standard PCI-to-PCI bridge OK
 0xEFF00000-0xFD6FFFFF PCI bus OK
 0xA0000-0xBFFFF PCI bus OK

```

0xA0000-0xBFFFF Standard VGA
Graphics Adapter OK
0xFD700000-0xFDFFFFFF PCI bus OK

0xFD9F0000-0xFD9F3FFF QLogic Fibre Channel
Adapter OK
0xFD800000-0xFD8FFFFFF QLogic Fibre Channel
Adapter OK
0xFD800000-0xFD8FFFFFF PCI standard PCI-to-
PCI bridge OK
0xFD2F0000-0xFD2F3FFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #3 OK

0xFD2C0000-0xFD2DFFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #3 OK

0xFD2A0000-0xFD2BFFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #3 OK

0xFC2C0000-0xFC2C00FF Standard Enhanced
PCI to USB Host Controller OK
0xFD300000-0xFD6FFFFFF PCI standard PCI-to-
PCI bridge OK
0xFD300000-0xFD6FFFFFF QLogic Fibre Channel
Adapter OK
0xFD6F0000-0xFD6F3FFF QLogic Fibre Channel
Adapter OK
0xFD500000-0xFD5FFFFFF QLogic Fibre Channel
Adapter OK
0xFD290000-0xFD293FFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #4 OK

0xFD260000-0xFD27FFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #4 OK

0xFD240000-0xFD25FFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #4 OK

0xFC290000-0xFC2900FF Standard Enhanced
PCI to USB Host Controller OK
0xFC3F0000-0xFC3FFFFFF Standard VGA
Graphics Adapter OK
0xC0000000-0xCFFFFFFF Motherboard
resources OK
0xFC6F0000-0xFC6F01FF HP ProLiant iLO 3
Management Controller (CORE) OK

0xFD4F0000-0xFD4F3FFF QLogic Fibre Channel
Adapter OK
0xFCFF0000-0xFCFF3FFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #2 OK

0xFCFC0000-0xFCFDFFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #2 OK

0xFCFA0000-0xFCFBFFFF HP NC551i Dual Port
FlexFabric 10Gb Network Adapter #2 OK

0xFC2E0000-0xFC2E0FFF Standard OpenHCD
USB Host Controller OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Manufacturer Description Status
File Version Size
Creation Date

```

```

c:\windows\system32\msg711.acm
Microsoft Corporation
OK
C:\Windows\system32\MSG711.AC
M 6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 5:18 PM
c:\windows\system32\imaadp32.acm
Microsoft Corporation
OK
C:\Windows\system32\IMAADP32.A
CM 6.1.7600.16385 21.50 KB
(22,016 bytes) 7/13/2009 5:18 PM

c:\windows\system32\msgsm32.acm
Microsoft Corporation
OK
C:\Windows\system32\MSGSM32.A
CM 6.1.7600.16385 28.50 KB
(29,184 bytes) 7/13/2009 5:18 PM

c:\windows\system32\msadp32.acm
Microsoft Corporation
OK
C:\Windows\system32\MSADP32.A
CM 6.1.7600.16385 23.50 KB
(24,064 bytes) 7/13/2009 5:18 PM

[Video Codecs]

CODEC Manufacturer Description Status
File Version Size
Creation Date

c:\windows\system32\msvidc32.dll
Microsoft Corporation
OK
C:\Windows\system32\MSVIDC32.D
LL 6.1.7600.16490 38.00 KB
(38,912 bytes) 3/3/2010 4:45 PM

c:\windows\system32\msrle32.dll
Microsoft Corporation
OK
C:\Windows\system32\MSRLE32.DL
L 6.1.7600.16490 16.00 KB
(16,384 bytes) 3/3/2010 4:45 PM

c:\windows\system32\msyuv.dll
Microsoft Corporation
OK
C:\Windows\system32\MSYUV.DLL
(25,088 bytes) 6.1.7600.16490 24.50 KB
3/3/2010 4:45 PM

c:\windows\system32\iyuv_32.dll
Microsoft Corporation
OK
C:\Windows\system32\IYUV_32.DL
L 6.1.7600.16490 53.00 KB
(54,272 bytes) 3/3/2010 4:45 PM

c:\windows\system32\tsbyuv.dll
Microsoft Corporation
OK
C:\Windows\system32\TSBYUV.DLL
(14,848 bytes) 6.1.7600.16490 14.50 KB
3/3/2010 4:45 PM

[CD-ROM]

Item Value

[Sound Device]

```

```

Item Value
[Display]

Item Value
Name Standard VGA Graphics Adapter

PNP Device ID
PCI\VEN_1002&DEV_515E&SUBSYS
_31FB103C&REV_02\4&2A2AE743&0&18A4

Adapter Type Not Available, (Standard display
types) compatible
Adapter Description Standard VGA
Graphics Adapter
Adapter RAM Not Available
Installed Drivers Not Available
Driver Version 6.1.7600.16385

INF File display.inf (vga section)
Color Planes Not Available
Color Table Entries Not Available
Resolution Not Available
Bits/Pixel Not Available
Memory Address 0xF0000000-
0xF7FFFFFF
I/O Port 0x00002000-0x00002FFF
Memory Address 0xFC3F0000-
0xFC3FFFFFF
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF

Driver
c:\windows\system32\drivers\vgapn
p.sys (6.1.7600.16385, 28.50 KB (29,184 bytes),
7/13/2009 4:38 PM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description USB Input Device
Name Enhanced (101- or 102-key)

Layout 00000409
PNP Device ID
USB\VID_0624&PID_0341&MI_00\6
&3648485&0&0000
Number of Function Keys 12
Driver
c:\windows\system32\drivers\hidus
b.sys (6.1.7600.16385, 29.50 KB (30,208 bytes),
7/13/2009 5:06 PM)

Description USB Input Device
Name Enhanced (101- or 102-key)

Layout 00000409
PNP Device ID
USB\VID_03F0&PID_7029&MI_00\7
&32D48B34&0&0000
Number of Function Keys 12
Driver
c:\windows\system32\drivers\hidus
b.sys (6.1.7600.16385, 29.50 KB (30,208 bytes),
7/13/2009 5:06 PM)

[Pointing Device]

```

Item Value
Hardware Type USB Input Device
Number of Buttons 0
Status OK
PNP Device ID
USB\VID_03F0&PID_7029&MI_01\7
&32D48B34&0&0001
Power Management Supported No
Double Click Threshold Not Available
Handedness Not Available
Driver
c:\windows\system32\drivers\hidus
b.sys (6.1.7600.16385, 29.50 KB (30,208 bytes),
7/13/2009 5:06 PM)

Hardware Type HID-compliant mouse
Number of Buttons 0
Status OK
PNP Device ID
HID\VID_0624&PID_0341&MI_01&
COL01\7&326A0541&0&0000
Power Management Supported No
Double Click Threshold Not Available
Handedness Not Available
Driver
c:\windows\system32\drivers\mouh
id.sys (6.1.7600.16385, 30.50 KB (31,232 bytes),
7/13/2009 5:00 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000000] WAN Miniport (SSTP)
Adapter TypeNot Available
Product Type WAN Miniport (SSTP)
Installed Yes
PNP Device ID
ROOT\MS_SSTPMINIORT\0000
Last Reset 1/7/2011 1:40 PM
Index 0
Service Name RasSstp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\rassst
p.sys (6.1.7600.16385, 82.00 KB (83,968 bytes),
7/13/2009 5:10 PM)
Name [00000001] WAN Miniport (IKEv2)
Adapter TypeNot Available
Product Type WAN Miniport (IKEv2)
Installed Yes

PNP Device ID
ROOT\MS_AGILEVPNMINIORT\00
Last Reset 1/7/2011 1:40 PM
Index 1
Service Name RasAgileVpn
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\agilev
pn.sys (6.1.7600.16385, 59.00 KB (60,416
bytes), 7/13/2009 5:10 PM)

Name [00000002] WAN Miniport (L2TP)
Adapter TypeNot Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID
ROOT\MS_L2TPMINIORT\0000

Last Reset 1/7/2011 1:40 PM
Index 2
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\rasl2t
p.sys (6.1.7600.16385, 127.00 KB (130,048
bytes), 7/13/2009 5:10 PM)

Name [00000003] WAN Miniport (PPTP)
Adapter TypeNot Available
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID
ROOT\MS_PPTPMINIORT\0000

Last Reset 1/7/2011 1:40 PM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\raspp
tp.sys (6.1.7600.16385, 109.00 KB (111,616
bytes), 7/13/2009 5:10 PM)

Name [00000004] WAN Miniport (PPPOE)
Adapter TypeNot Available
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID
ROOT\MS_PPPOEMINIORT\0000
Last Reset 1/7/2011 1:40 PM

Service Name4 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 Not Available
Driver
c:\windows\system32\drivers\raspp
poe.sys (6.1.7600.16385, 90.50 KB (92,672
bytes), 7/13/2009 5:10 PM)

Name [00000005] WAN Miniport (IPv6)
Adapter TypeNot Available
Product Type WAN Miniport (IPv6)
Installed Yes
PNP Device ID
ROOT\MS_NDISWANIPV6\0000

Last Reset 1/7/2011 1:40 PM
Index 5
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\ndisw
an.sys (6.1.7600.16385, 160.50 KB (164,352
bytes), 7/13/2009 5:10 PM)

Name [00000006] WAN Miniport (Network
Monitor)
Adapter TypeNot Available
Product Type WAN Miniport (Network Monitor)

Installed Yes
PNP Device ID
ROOT\MS_NDISWANBH\0000
Last Reset 1/7/2011 1:40 PM
Index 6
Service Name NdisWan

IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\ndisw
an.sys (6.1.7600.16385, 160.50 KB (164,352
bytes), 7/13/2009 5:10 PM)

Name [00000007] Broadcom BCM5709C
NetXtreme II GigE (NDIS VBD Client)
Adapter TypeNot Available
Product TypeBroadcom BCM5709C NetXtreme II
GigE (NDIS VBD Client)
Installed Yes
PNP Device ID Not Available
Last Reset 1/7/2011 1:40 PM
Index 7
Service Name l2nd
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] Microsoft 6to4 Adapter

Adapter Type Tunnel
 Product Type Microsoft 6to4 Adapter
 Installed Yes
 PNP Device ID ROOT*6TO4MP\0001

Last Reset 1/7/2011 1:40 PM
 Index 8
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

Name [00000009] WAN Miniport (IP)

Adapter Type Not Available
 Product Type WAN Miniport (IP)
 Installed Yes
 PNP Device ID ROOT\MS_NDISWANIP\0000

Last Reset 1/7/2011 1:40 PM
 Index 9
 Service Name NdisWan
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\ndisw
 an.sys (6.1.7600.16385, 160.50 KB (164,352
 bytes), 7/13/2009 5:10 PM)

Name [00000010] Broadcom BCM5709C
 NetXtreme II GigE (NDIS VBD Client)
 Adapter Type Not Available
 Product Type Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 10
 Service Name i2nd
 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] RAS Async Adapter

Product Type RAS Async Adapter (WAN)
 Installed Yes
 PNP Device ID SW\{EEAB7790-C514-
 11D1-B42B-00805FC1270E}\ASYNCMAC
 Last Reset 1/7/2011 1:40 PM
 Index 11
 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 20:41:53:59:4E:FF
 Driver c:\windows\system32\drivers\async
 mac.sys (6.1.7600.16385, 22.50 KB (23,040
 bytes), 7/13/2009 5:10 PM)

Name [00000012] Microsoft ISATAP
 Adapter
 Adapter Type Tunnel
 Product Type Microsoft ISATAP Adapter
 Installed Yes
 PNP Device ID ROOT*ISATAP\0001

Last Reset 1/7/2011 1:40 PM
 Index 12
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

Name [00000013] Broadcom BCM5709C
 NetXtreme II GigE (NDIS VBD Client)
 Adapter Type Not Available
 Product Type Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 13
 Service Name i2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000014] Microsoft ISATAP
 Adapter
 Adapter Type Tunnel
 Product Type Microsoft ISATAP Adapter
 Installed Yes
 PNP Device ID ROOT*ISATAP\0002

Last Reset 1/7/2011 1:40 PM
 Index 14
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available

DHCP Enabled Not Available No
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

Name [00000015] Broadcom BCM5709C
 NetXtreme II GigE (NDIS VBD Client)
 Adapter Type Not Available
 Product Type Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 15
 Service Name i2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000016] Microsoft ISATAP
 Adapter
 Adapter Type Tunnel
 Product Type Microsoft ISATAP Adapter
 Installed Yes
 PNP Device ID ROOT*ISATAP\0003

Last Reset 1/7/2011 1:40 PM
 Index 16
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

Name [00000017] Microsoft 6to4 Adapter

Adapter Type Tunnel
 Product Type Microsoft 6to4 Adapter
 Installed Yes
 PNP Device ID ROOT*6TO4MP\0000

Last Reset 1/7/2011 1:40 PM
 Index 17
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

Name [00000018] HP NC550SFP Dual Port 10GbE Server Adapter
 Adapter TypeNot Available
 Product TypeHP NC550SFP Dual Port 10GbE Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 18
 Service Name be2net
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000019] HP NC550SFP Dual Port 10GbE Server Adapter
 Adapter TypeNot Available
 Product TypeHP NC550SFP Dual Port 10GbE Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 19
 Service Name be2net
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000020] Microsoft ISATAP Adapter
 Adapter TypeTunnel
 Product TypeMicrosoft ISATAP Adapter
 Installed Yes
 PNP Device ID ROOT*ISATAP\0004
 Last Reset 1/7/2011 1:40 PM
 Index 20
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver c:\windows\system32\drivers\tunnel.sys (6.1.7600.16385, 122.50 KB (125,440 bytes), 7/13/2009 5:09 PM)

Name [00000021] Intel(R) PRO/1000 PT Dual Port Server Adapter
 Adapter TypeNot Available
 Product TypeIntel(R) PRO/1000 PT Dual Port Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 21
 Service Name e1express
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes

DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000022] Intel(R) PRO/1000 PT Dual Port Server Adapter
 Adapter TypeNot Available
 Product TypeIntel(R) PRO/1000 PT Dual Port Server Adapter
 Installed Yes
 PNP Device ID Not Available
 Last Reset 1/7/2011 1:40 PM
 Index 22
 Service Name e1express
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000023] HP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Adapter TypeEthernet 802.3
 Product TypeHP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Installed Yes
 PNP Device ID PCI\VEN_19A2&DEV_0700&SUBSYS_3314103C&REV_02\4&3636F4A7&0&0110

Last Reset 1/7/2011 1:40 PM
 Index 23
 Service Name be2net
 IP Address 192.168.2.3
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 18:A9:05:5D:C4:DE
 Memory Address 0xFCF90000-0xFCF93FFF
 Memory Address 0xFCF60000-0xFCF7FFFF
 Memory Address 0xFCF40000-0xFCF5FFFF
 IRQ Channel IRQ 4294967070
 IRQ Channel IRQ 4294967069
 IRQ Channel IRQ 4294967068
 IRQ Channel IRQ 4294967067
 IRQ Channel IRQ 4294967066
 IRQ Channel IRQ 4294967065
 IRQ Channel IRQ 4294967064
 IRQ Channel IRQ 4294967063
 IRQ Channel IRQ 4294967062
 IRQ Channel IRQ 4294967061
 IRQ Channel IRQ 4294967060
 IRQ Channel IRQ 4294967059
 Driver c:\windows\system32\drivers\be2nd62.sys (2.101.374.3, 332.55 KB (340,528 bytes), 3/24/2010 1:25 PM)

Name [00000024] HP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Adapter TypeEthernet 802.3
 Product TypeHP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Installed Yes

PNP Device ID PCI\VEN_19A2&DEV_0700&SUBSYS_3314103C&REV_02\4&3636F4A7&0&0010
 Last Reset 1/7/2011 1:40 PM
 Index 24
 Service Name be2net
 IP Address 192.168.2.1
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 18:A9:05:5D:C4:DA
 Memory Address 0xFCFF0000-0xFCFF3FFF
 Memory Address 0xFCFC0000-0xFCFDFFFF
 Memory Address 0xFCFA0000-0xFCFBFFFF
 IRQ Channel IRQ 4294967058
 IRQ Channel IRQ 4294967057
 IRQ Channel IRQ 4294967056
 IRQ Channel IRQ 4294967055
 IRQ Channel IRQ 4294967054
 IRQ Channel IRQ 4294967053
 IRQ Channel IRQ 4294967052
 IRQ Channel IRQ 4294967051
 IRQ Channel IRQ 4294967050
 IRQ Channel IRQ 4294967049
 IRQ Channel IRQ 4294967048
 IRQ Channel IRQ 4294967047
 Driver c:\windows\system32\drivers\be2nd62.sys (2.101.374.3, 332.55 KB (340,528 bytes), 3/24/2010 1:25 PM)

Name [00000025] HP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Adapter TypeEthernet 802.3
 Product TypeHP NC551i Dual Port FlexFabric 10Gb Network Adapter
 Installed Yes
 PNP Device ID PCI\VEN_19A2&DEV_0700&SUBSYS_3314103C&REV_02\4&2089B542&0&0018
 Last Reset 1/7/2011 1:40 PM
 Index 25
 Service Name be2net
 IP Address 192.168.2.2
 IP Subnet 255.255.0.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 18:A9:05:5D:C4:E2
 Memory Address 0xFD2F0000-0xFD2F3FFF
 Memory Address 0xFD2C0000-0xFD2DFFFF
 Memory Address 0xFD2A0000-0xFD2BFFFF
 IRQ Channel IRQ 4294967082
 IRQ Channel IRQ 4294967081
 IRQ Channel IRQ 4294967080
 IRQ Channel IRQ 4294967079
 IRQ Channel IRQ 4294967078
 IRQ Channel IRQ 4294967077
 IRQ Channel IRQ 4294967076
 IRQ Channel IRQ 4294967075
 IRQ Channel IRQ 4294967074
 IRQ Channel IRQ 4294967073
 IRQ Channel IRQ 4294967072

IRQ Channel IRQ 4294967071
Driver
c:\windows\system32\drivers\be2n
d62.sys (2.101.374.3, 332.55 KB (340,528
bytes), 3/24/2010 1:25 PM)

Name [00000026] HP NC551i Dual Port
FlexFabric 10Gb Network Adapter
Adapter Type Ethernet 802.3
Product Type HP NC551i Dual Port FlexFabric
10Gb Network Adapter
Installed Yes
PNP Device ID
PCI\VEN_19A2&DEV_0700&SUBSYS
_3314103C&REV_02\4&2089B542&0&0118

Last Reset 1/7/2011 1:40 PM
Index 26
Service Name be2net
IP Address 15.1.101.57,
fe80::c8ea:dd08:54fe:623
IP Subnet 255.255.0.0, 64
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server 15.1.101.1
DHCP Lease Expires 1/19/2011 1:40 PM
DHCP Lease Obtained 1/11/2011 1:40 PM

MAC Address 18:A9:05:5D:C4:E6
Memory Address 0xFD290000-
0xFD293FFF
Memory Address 0xFD260000-
0xFD27FFFF
Memory Address 0xFD240000-
0xFD25FFFF

IRQ Channel IRQ 4294967094
IRQ Channel IRQ 4294967093
IRQ Channel IRQ 4294967092
IRQ Channel IRQ 4294967091
IRQ Channel IRQ 4294967090
IRQ Channel IRQ 4294967089
IRQ Channel IRQ 4294967088
IRQ Channel IRQ 4294967087
IRQ Channel IRQ 4294967086
IRQ Channel IRQ 4294967085
IRQ Channel IRQ 4294967084
IRQ Channel IRQ 4294967083

Driver
c:\windows\system32\drivers\be2n
d62.sys (2.101.374.3, 332.55 KB (340,528
bytes), 3/24/2010 1:25 PM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.99 KB (65,527 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	MSAFD Tcpip [TCP/IPv6]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	28 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	28 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/IPv6]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	28 bytes
Maximum Message Size	63.99 KB (65,527 bytes)
Message Oriented	Yes
Minimum Address Size	28 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 TCPv6 Service Provider
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	28 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	28 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	Yes
Supports Graceful Closing	Yes

Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	RSVP TCP Service Provider
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	RSVP UDPv6 Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	28 bytes
Maximum Message Size	63.99 KB (65,527 bytes)
Message Oriented	Yes
Minimum Address Size	28 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	Yes
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.99 KB (65,527 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

[WinSock]

Item	Value
File	c:\windows\syswow64\wsoc32.dll
Size	15.00 KB (15,360 bytes)
Version	6.1.7600.16385
File	c:\windows\system32\wsoc32.dll
Size	18.00 KB (18,432 bytes)
Version	6.1.7600.16385

[Ports]

[Serial]

Item Value
Name Communications Port (COM2)

Status OK
PNP Device ID ACPI\PNP0501\1

Maximum Input Buffer Size0
Maximum Output Buffer Size No

Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes
Settable Parity Check Yes
Settable Stop Bits Yes
Settable RLSD Yes
Supports RLSD Yes
Supports 16 Bit Mode No
Supports Special Characters No

Baud Rate 9600
Bits/Byte 8
Stop Bits 1
Parity None
Busy No

Abort Read/Write on Error No
Binary Mode Enabled Yes
Continue XMit on XOff No
CTS Outflow Control No
Discard NULL Bytes No
DSR Outflow Control 0
DSR Sensitivity 0
DTR Flow Control Type Enable
EOF Character 0
Error Replace Character 0
Error Replacement Enabled No

Event Character 0
Parity Check Enabled No
RTS Flow Control Type Enable
XOff Character 19
XOffXMit Threshold 512
XOn Character 17
XOnXMit Threshold 2048
XOnXOff InFlow Control 0
XOnXOff OutFlow Control 0
IRQ Channel IRQ 3
I/O Port 0x000002F8-0x000002FF
Driver

c:\windows\system32\drivers\serial.sys (6.1.7600.16385, 92.00 KB (94,208 bytes), 7/13/2009 5:00 PM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS

Size 68.23 GB (73,265,049,600 bytes)
Free Space 20.45 GB (21,960,138,752 bytes)

Volume Name
Volume Serial Number 929E7AFC

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 S:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 100.00 MB (104,853,504 bytes)

Free Space 71.84 MB (75,333,632 bytes)

Volume Name System Reserved

Volume Serial Number 029D53C1

Drive T:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 299.23 GB (321,298,362,368 bytes)

Free Space 276.44 GB (296,825,540,608 bytes)

Volume Name New Volume
Volume Serial Number A24D2A05

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)
Model HP LOGICAL VOLUME SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 4
Sectors/Track 32
Size 68.33 GB (73,372,631,040 bytes)

Total Cylinders 17,562
Total Sectors 143,305,920
Total Tracks 4,478,310
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 100.00 MB (104,857,600 bytes)

Partition Starting Offset 1,048,576 bytes

Partition Disk #0, Partition #1
Partition Size 68.23 GB (73,265,053,696 bytes)

Partition Starting Offset 105,906,176 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512 Yes
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 15
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
Total Sectors 1,575,173,250
Total Tracks 25,002,750
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 16
SCSI Port 8
SCSI Target ID 0
Sectors/Track 63
Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
Total Sectors 1,575,173,250
Total Tracks 25,002,750
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 2
SCSI Port 8
SCSI Target ID 1
Sectors/Track 63
Size 2.27 TB (2,497,951,733,760 bytes)

Total Cylinders 303,692
Total Sectors 4,878,811,980
Total Tracks 77,441,460
Tracks/Cylinder 255
Partition Disk #18, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #18, Partition #1
Partition Size 2.27 TB (2,497,820,523,008 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)

Model HP MSA2324fc SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 8
 SCSI Target ID 2
 Sectors/Track 63
 Size 5.00 TB (5,495,498,749,440 bytes)

Total Cylinders 668,123
 Total Sectors 10,733,395,995
 Total Tracks 170,371,365
 Tracks/Cylinder 255
 Partition Disk #19, Partition #0
 Partition Size 5.00 TB (5,495,364,517,888 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 7
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 9
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #22, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)

Model HP MSA2324fc SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 5
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #1, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 5
 SCSI Target ID 1
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 751.10 GB (806,492,307,456 bytes)

Partition Starting Offset 1,048,576 bytes

Description Disk drive
 Manufacturer (Standard disk drives)

Model HP MSA2324fc SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 5
 SCSI Target ID 1
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 7
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #12, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)

Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 7
 SCSI Target ID 1
 Sectors/Track 63
 Size 2.27 TB (2,497,951,733,760 bytes)

Total Cylinders 303,692
 Total Sectors 4,878,811,980
 Total Tracks 77,441,460
 Tracks/Cylinder 255
 Partition Disk #13, Partition #0
 Partition Size 1.00 MB (1,048,576 bytes)
 Partition Starting Offset 17,408 bytes
 Partition Disk #13, Partition #1
 Partition Size 2.27 TB (2,497,820,523,008 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 7
 SCSI Target ID 2
 Sectors/Track 63
 Size 5.00 TB (5,495,498,749,440 bytes)

Total Cylinders 668,123
 Total Sectors 10,733,395,995
 Total Tracks 170,371,365
 Tracks/Cylinder 255
 Partition Disk #14, Partition #0
 Partition Size 5.00 TB (5,495,364,517,888 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 9
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 10
 SCSI Port 6
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 11
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 12
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 13
 SCSI Port 10
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model HP MSA2324fc SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus 0
 SCSI Logical Unit 14
 SCSI Port 10
 SCSI Target ID 0
 Sectors/Track 63
 Size 751.10 GB (806,488,704,000 bytes)

Total Cylinders 98,050
 Total Sectors 1,575,173,250
 Total Tracks 25,002,750
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 750.98 GB (806,359,138,304 bytes)

Partition Starting Offset 135,266,304 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 8
 SCSI Target ID 3
 Sectors/Track 63
 Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866
 Total Sectors 3,355,432,290
 Total Tracks 53,260,830
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0
 Partition Size 1.00 MB (1,048,576 bytes)
 Partition Starting Offset 17,408 bytes
 Partition Disk #20, Partition #1
 Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 9
SCSI Target ID 1
Sectors/Track 63
Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866
Total Sectors 3,355,432,290
Total Tracks 53,260,830
Tracks/Cylinder 255
Partition Disk #23, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #23, Partition #1
Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 2
Sectors/Track 63
Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866
Total Sectors 3,355,432,290
Total Tracks 53,260,830
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #5, Partition #1
Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 3
Sectors/Track 63
Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866

Total Tracks 53,260,830
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #15, Partition #1
Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 2
Sectors/Track 63
Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866
Total Sectors 3,355,432,290
Total Tracks 53,260,830
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #10, Partition #1
Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model VIOLIN 1010 SAN SCSI Disk Device

Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 2
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 10
SCSI Target ID 1
Sectors/Track 63
Size 1.56 TB (1,717,981,332,480 bytes)

Total Cylinders 208,866
Total Sectors 3,355,432,290
Total Tracks 53,260,830
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 1.00 MB (1,048,576 bytes)
Partition Starting Offset 17,408 bytes
Partition Disk #26, Partition #1
Partition Size 1.56 TB (1,717,852,666,368 bytes)

Partition Starting Offset 134,235,136 bytes

[SCSI]

Item Value
Name QLogic Fibre Channel Adapter

Manufacturer QLogic
Status OK

PNP Device ID
PCI\VEN_1077&DEV_2532&SUBSYS
_3261103C&REV_02\4&32FD6DD9&80&0010
Memory Address 0xFDF0000-
0xFDF3FFF

Memory Address 0xFDE00000-
0xFDEFFFFF
IRQ Channel IRQ 4294967222
IRQ Channel IRQ 4294967221
IRQ Channel IRQ 4294967220
IRQ Channel IRQ 4294967219
IRQ Channel IRQ 4294967218
IRQ Channel IRQ 4294967217
IRQ Channel IRQ 4294967216
IRQ Channel IRQ 4294967215
IRQ Channel IRQ 4294967214
IRQ Channel IRQ 4294967213
IRQ Channel IRQ 4294967212
IRQ Channel IRQ 4294967211
IRQ Channel IRQ 4294967210
IRQ Channel IRQ 4294967209
IRQ Channel IRQ 4294967208
IRQ Channel IRQ 4294967207
IRQ Channel IRQ 4294967206
IRQ Channel IRQ 4294967205
IRQ Channel IRQ 4294967204
IRQ Channel IRQ 4294967203
IRQ Channel IRQ 4294967202
IRQ Channel IRQ 4294967201
IRQ Channel IRQ 4294967200
IRQ Channel IRQ 4294967199
IRQ Channel IRQ 4294967198
IRQ Channel IRQ 4294967197
IRQ Channel IRQ 4294967196
IRQ Channel IRQ 4294967195
IRQ Channel IRQ 4294967194
IRQ Channel IRQ 4294967193
IRQ Channel IRQ 4294967192
IRQ Channel IRQ 4294967191

Driver
c:\windows\system32\drivers\ql230
0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
9/9/2010 8:01 AM)

Name Smart Array P410i Controller

Manufacturer Hewlett-Packard Company
Status OK
PNP Device ID
PCI\VEN_103C&DEV_323A&SUBSYS
_3245103C&REV_01\4&1AB23AD6&80&0060

Memory Address 0xFC800000-
0xFCBFFFFF
Memory Address 0xFC7F0000-
0xFC7F0FFF
IRQ Channel IRQ 4294967294
IRQ Channel IRQ 4294967293
IRQ Channel IRQ 4294967292
IRQ Channel IRQ 4294967291
IRQ Channel IRQ 4294967290
IRQ Channel IRQ 4294967289
IRQ Channel IRQ 4294967288
IRQ Channel IRQ 4294967287

Driver
c:\windows\system32\drivers\hpcis
ss2.sys (6.20.0.64, 153.10 KB (156,776 bytes),
4/9/2010 10:05 AM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic
Status OK

PNP Device ID
 PCI\VEN_1077&DEV_2532&SUBSYS
 _3261103C&REV_02\4&32FD6DD9&0&0110

Memory Address 0xFDDF0000-
 0xFDDF3FFF
 Memory Address 0xFDC00000-
 0xFDCFFFFF
 IRQ Channel IRQ 4294967190
 IRQ Channel IRQ 4294967189
 IRQ Channel IRQ 4294967188
 IRQ Channel IRQ 4294967187
 IRQ Channel IRQ 4294967186
 IRQ Channel IRQ 4294967185
 IRQ Channel IRQ 4294967184
 IRQ Channel IRQ 4294967183
 IRQ Channel IRQ 4294967182
 IRQ Channel IRQ 4294967181
 IRQ Channel IRQ 4294967180
 IRQ Channel IRQ 4294967179
 IRQ Channel IRQ 4294967178
 IRQ Channel IRQ 4294967177
 IRQ Channel IRQ 4294967176
 IRQ Channel IRQ 4294967175
 IRQ Channel IRQ 4294967174
 IRQ Channel IRQ 4294967173
 IRQ Channel IRQ 4294967172
 IRQ Channel IRQ 4294967171
 IRQ Channel IRQ 4294967170
 IRQ Channel IRQ 4294967169
 IRQ Channel IRQ 4294967168
 IRQ Channel IRQ 4294967167
 IRQ Channel IRQ 4294967166
 IRQ Channel IRQ 4294967165
 IRQ Channel IRQ 4294967164
 IRQ Channel IRQ 4294967163
 IRQ Channel IRQ 4294967162
 IRQ Channel IRQ 4294967161
 IRQ Channel IRQ 4294967160
 IRQ Channel IRQ 4294967159
 Driver
 c:\windows\system32\drivers\ql230
 0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
 9/9/2010 8:01 AM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2532&SUBSYS
 _3261103C&REV_02\4&1DEBC1A4&0&0060

Memory Address 0xFDBF0000-
 0xFDBF3FFF
 Memory Address 0xFDA00000-
 0xFDAFFFFF
 IRQ Channel IRQ 4294967158
 IRQ Channel IRQ 4294967157
 IRQ Channel IRQ 4294967156
 IRQ Channel IRQ 4294967155
 IRQ Channel IRQ 4294967154
 IRQ Channel IRQ 4294967153
 IRQ Channel IRQ 4294967152
 IRQ Channel IRQ 4294967151
 IRQ Channel IRQ 4294967150
 IRQ Channel IRQ 4294967149
 IRQ Channel IRQ 4294967148
 IRQ Channel IRQ 4294967147
 IRQ Channel IRQ 4294967146
 IRQ Channel IRQ 4294967145
 IRQ Channel IRQ 4294967144
 IRQ Channel IRQ 4294967143
 IRQ Channel IRQ 4294967142
 IRQ Channel IRQ 4294967141

IRQ Channel IRQ 4294967140
 IRQ Channel IRQ 4294967138
 IRQ Channel IRQ 4294967137
 IRQ Channel IRQ 4294967136
 IRQ Channel IRQ 4294967135
 IRQ Channel IRQ 4294967134
 IRQ Channel IRQ 4294967133
 IRQ Channel IRQ 4294967132
 IRQ Channel IRQ 4294967131
 IRQ Channel IRQ 4294967130
 IRQ Channel IRQ 4294967129
 IRQ Channel IRQ 4294967128
 IRQ Channel IRQ 4294967127
 Driver
 c:\windows\system32\drivers\ql230
 0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
 9/9/2010 8:01 AM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2532&SUBSYS
 _3261103C&REV_02\4&1DEBC1A4&0&0160

Memory Address 0xFD9F0000-
 0xFD9F3FFF
 Memory Address 0xFD800000-
 0xFD8FFFFF
 IRQ Channel IRQ 4294967126
 IRQ Channel IRQ 4294967125
 IRQ Channel IRQ 4294967124
 IRQ Channel IRQ 4294967123
 IRQ Channel IRQ 4294967122
 IRQ Channel IRQ 4294967121
 IRQ Channel IRQ 4294967120
 IRQ Channel IRQ 4294967119
 IRQ Channel IRQ 4294967118
 IRQ Channel IRQ 4294967117
 IRQ Channel IRQ 4294967116
 IRQ Channel IRQ 4294967115
 IRQ Channel IRQ 4294967114
 IRQ Channel IRQ 4294967113
 IRQ Channel IRQ 4294967112
 IRQ Channel IRQ 4294967111
 IRQ Channel IRQ 4294967110
 IRQ Channel IRQ 4294967109
 IRQ Channel IRQ 4294967108
 IRQ Channel IRQ 4294967107
 IRQ Channel IRQ 4294967106
 IRQ Channel IRQ 4294967105
 IRQ Channel IRQ 4294967104
 IRQ Channel IRQ 4294967103
 IRQ Channel IRQ 4294967102
 IRQ Channel IRQ 4294967101
 IRQ Channel IRQ 4294967100
 IRQ Channel IRQ 4294967099
 IRQ Channel IRQ 4294967098
 IRQ Channel IRQ 4294967097
 IRQ Channel IRQ 4294967096
 IRQ Channel IRQ 4294967095
 Driver
 c:\windows\system32\drivers\ql230
 0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
 9/9/2010 8:01 AM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2532&SUBSYS
 _3261103C&REV_02\4&25A804AF&0&0058

Memory Address 0xFD6F0000-
 0xFD6F3FFF
 Memory Address 0xFD500000-
 0xFD5FFFFF
 IRQ Channel IRQ 4294967286
 IRQ Channel IRQ 4294967285
 IRQ Channel IRQ 4294967284
 IRQ Channel IRQ 4294967283
 IRQ Channel IRQ 4294967282
 IRQ Channel IRQ 4294967281
 IRQ Channel IRQ 4294967280
 IRQ Channel IRQ 4294967279
 IRQ Channel IRQ 4294967278
 IRQ Channel IRQ 4294967277
 IRQ Channel IRQ 4294967276
 IRQ Channel IRQ 4294967275
 IRQ Channel IRQ 4294967274
 IRQ Channel IRQ 4294967273
 IRQ Channel IRQ 4294967272
 IRQ Channel IRQ 4294967271
 IRQ Channel IRQ 4294967270
 IRQ Channel IRQ 4294967269
 IRQ Channel IRQ 4294967268
 IRQ Channel IRQ 4294967267
 IRQ Channel IRQ 4294967266
 IRQ Channel IRQ 4294967265
 IRQ Channel IRQ 4294967264
 IRQ Channel IRQ 4294967263
 IRQ Channel IRQ 4294967262
 IRQ Channel IRQ 4294967261
 IRQ Channel IRQ 4294967260
 IRQ Channel IRQ 4294967259
 IRQ Channel IRQ 4294967258
 IRQ Channel IRQ 4294967257
 IRQ Channel IRQ 4294967256
 IRQ Channel IRQ 4294967255
 Driver
 c:\windows\system32\drivers\ql230
 0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
 9/9/2010 8:01 AM)

Name QLogic Fibre Channel Adapter

Manufacturer QLogic
 Status OK
 PNP Device ID
 PCI\VEN_1077&DEV_2532&SUBSYS
 _3261103C&REV_02\4&25A804AF&0&0158

Memory Address 0xFD4F0000-
 0xFD4F3FFF
 Memory Address 0xFD300000-
 0xFD6FFFFF
 IRQ Channel IRQ 4294967254
 IRQ Channel IRQ 4294967253
 IRQ Channel IRQ 4294967252
 IRQ Channel IRQ 4294967251
 IRQ Channel IRQ 4294967250
 IRQ Channel IRQ 4294967249
 IRQ Channel IRQ 4294967248
 IRQ Channel IRQ 4294967247
 IRQ Channel IRQ 4294967246
 IRQ Channel IRQ 4294967245
 IRQ Channel IRQ 4294967244
 IRQ Channel IRQ 4294967243
 IRQ Channel IRQ 4294967242
 IRQ Channel IRQ 4294967241
 IRQ Channel IRQ 4294967240
 IRQ Channel IRQ 4294967239
 IRQ Channel IRQ 4294967238
 IRQ Channel IRQ 4294967237
 IRQ Channel IRQ 4294967236
 IRQ Channel IRQ 4294967235
 IRQ Channel IRQ 4294967234
 IRQ Channel IRQ 4294967233
 IRQ Channel IRQ 4294967232

IRQ Channel IRQ 4294967231
 IRQ Channel IRQ 4294967230
 IRQ Channel IRQ 4294967229
 IRQ Channel IRQ 4294967228
 IRQ Channel IRQ 4294967227
 IRQ Channel IRQ 4294967226
 IRQ Channel IRQ 4294967225
 IRQ Channel IRQ 4294967224
 IRQ Channel IRQ 4294967223

Driver
 c:\windows\system32\drivers\ql230
 0.sys (9.1.8.27, 1.14 MB (1,196,656 bytes),
 9/9/2010 8:01 AM)

[IDE]

Item Value
 Name Standard Dual Channel PCI IDE
 Controller
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK
 PNP Device ID
 PCI\VEN_1002&DEV_439C&SUBSYS
 _1773103C&REV_00\3&3097523A&0&A1

I/O Port 0x00000500-0x0000050F
 Driver
 c:\windows\system32\drivers\pciide
 .sys (6.1.7600.16385, 12.06 KB (12,352 bytes),
 7/13/2009 4:19 PM)

Name Standard Dual Channel PCI IDE
 Controller
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK
 PNP Device ID
 PCI\VEN_1002&DEV_4390&SUBSYS
 _176E103C&REV_00\3&3097523A&0&88

I/O Port 0x00001000-0x00001007
 I/O Port 0x00001008-0x0000100B
 I/O Port 0x00001010-0x00001017
 I/O Port 0x00001018-0x0000101B
 I/O Port 0x00001020-0x0000102F
 Memory Address 0xFC2F0000-
 0xFC2F03FF
 IRQ Channel IRQ 16
 Driver
 c:\windows\system32\drivers\pciide
 .sys (6.1.7600.16385, 12.06 KB (12,352 bytes),
 7/13/2009 4:19 PM)

Name ATA Channel 0
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK
 PNP Device ID
 PCI\IDE\IDECHANNEL\4&8120232&
 0&0
 I/O Port 0x000001F0-0x000001F7
 I/O Port 0x000003F6-0x000003F6
 IRQ Channel IRQ 14
 Driver
 c:\windows\system32\drivers\atapi.
 sys (6.1.7600.16385, 23.56 KB (24,128 bytes),
 7/13/2009 4:19 PM)

Name ATA Channel 1
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK

PNP Device ID
 PCI\IDE\IDECHANNEL\4&8120232&
 0&0
 I/O Port 0x00000170-0x00000177
 I/O Port 0x00000376-0x00000376
 IRQ Channel IRQ 15
 Driver
 c:\windows\system32\drivers\atapi.
 sys (6.1.7600.16385, 23.56 KB (24,128 bytes),
 7/13/2009 4:19 PM)

Name ATA Channel 0
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK
 PNP Device ID
 PCI\IDE\IDECHANNEL\4&C1BABAC&
 0&0
 Driver
 c:\windows\system32\drivers\atapi.
 sys (6.1.7600.16385, 23.56 KB (24,128 bytes),
 7/13/2009 4:19 PM)

Name ATA Channel 1
 Manufacturer (Standard IDE ATA/ATAPI
 controllers)
 Status OK
 PNP Device ID
 PCI\IDE\IDECHANNEL\4&C1BABAC&
 0&1
 Driver
 c:\windows\system32\drivers\atapi.
 sys (6.1.7600.16385, 23.56 KB (24,128 bytes),
 7/13/2009 4:19 PM)

[Printing]

Can't Collect Information

[Problem Devices]

Device Code	PNP Device ID	Error
Mass Storage Controller	PCI\VEN_19A2&DEV_0702&SUBSYS _3314103C&REV_02\4&2089B542&0&0218	The drivers for this device are not installed.
Standard PS/2 Keyboard	ACPI\PNP0303\4&9333F3&0	This device is not present, is not working properly, or does not have all its drivers installed.
Mass Storage Controller	PCI\VEN_19A2&DEV_0702&SUBSYS _3314103C&REV_02\4&2089B542&0&0318	The drivers for this device are not installed.
Mass Storage Controller	PCI\VEN_19A2&DEV_0702&SUBSYS _3314103C&REV_02\4&3636F4A7&0&0210	The drivers for this device are not installed.
Mass Storage Controller	PCI\VEN_19A2&DEV_0702&SUBSYS _3314103C&REV_02\4&3636F4A7&0&0310	The drivers for this device are not installed.
PS/2 Compatible Mouse	ACPI\PNP0F13\4&9333F3&0	This device is not present, is not working properly, or does not have all its drivers installed.

[USB]

Device PNP Device ID
 Standard OpenHCD USB Host Controller
 PCI\VEN_1002&DEV_4397&SUBSYS
 _176F103C&REV_00\3&3097523A&0&898

Standard OpenHCD USB Host Controller
 PCI\VEN_1002&DEV_4398&SUBSYS
 _1770103C&REV_00\3&3097523A&0&891

Standard OpenHCD USB Host Controller
 PCI\VEN_1002&DEV_4396&SUBSYS
 _1770103C&REV_00\3&3097523A&0&899

Standard Universal PCI to USB Host Controller
 PCI\VEN_103C&DEV_3300&SUBSYS
 _3309103C&REV_01\4&2F88CDA0&0&0450

Standard Enhanced PCI to USB Host Controller
 PCI\VEN_1002&DEV_4396&SUBSYS
 _1771103C&REV_00\3&3097523A&0&892

Standard Enhanced PCI to USB Host Controller
 PCI\VEN_1002&DEV_4396&SUBSYS
 _1771103C&REV_00\3&3097523A&0&89A

Standard OpenHCD USB Host Controller
 PCI\VEN_1002&DEV_4397&SUBSYS
 _176F103C&REV_00\3&3097523A&0&890

[Software Environment]

[System Drivers]

Name	Description	File	Type
	Started	Start Mode	State
	Status	Error Control	Accept
Pause	Accept Stop		
1394ohci Controller	1394 OHCI Compliant Host		
ohci.sys	c:\windows\system32\drivers\1394 Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
acpi	Microsoft ACPI Driver		
ys	c:\windows\system32\drivers\acpi.s Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
acpipmi	ACPI Power Meter Driver		
mi.sys	c:\windows\system32\drivers\acpi Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
adp94xx	adp94xx		
4xx.sys	c:\windows\system32\drivers\adp9 Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
adpahci	adpahci		
hci.sys	c:\windows\system32\drivers\adpa Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
adpu320	adpu320		
320.sys	c:\windows\system32\drivers\adpu Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	

afd	Ancillary Function Driver for		
Winsock			
	c:\windows\system32\drivers\afd.sys		
s	Kernel Driver Yes	System	
	Running OK	Normal	
	No Yes		
agp440	Intel AGP Bus Filter		
	c:\windows\system32\drivers\agp4		
40.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
aliide	aliide		
	c:\windows\system32\drivers\aliide.		
sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
amdide	amdide		
	c:\windows\system32\drivers\amdi		
de.sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
amdk8	AMD K8 Processor Driver		
	c:\windows\system32\drivers\amdk		
8.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
amdppm	AMD Processor Driver		
	c:\windows\system32\drivers\amdp		
pm.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
amdsata	amdsata		
	c:\windows\system32\drivers\amds		
ata.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
amdsbs	amdsbs		
	c:\windows\system32\drivers\amds		
bs.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
amdxata	amdxata		
	c:\windows\system32\drivers\amdx		
ata.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
appid	AppID Driver		
	c:\windows\system32\drivers\appid		
.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
arc	arc		
	c:\windows\system32\drivers\arc.sys		
s	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
arcsas	arcsas		
	c:\windows\system32\drivers\arcsa		
s.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
asynmac	RAS Asynchronous Media Driver		
	c:\windows\system32\drivers\asyn		
mac.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
atapi	IDE Channel		
	c:\windows\system32\drivers\atapi.		
sys	Kernel Driver Yes	Boot	
	Running OK	Critical	
	No Yes		

b06bdrv	Broadcom NetXtreme II VBD		
	c:\windows\system32\drivers\bxvbd		
a.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
b57nd60a	Broadcom NetXtreme Gigabit		
	Ethernet - NDIS 6.0		
	c:\windows\system32\drivers\b57n		
d60a.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
be2net	ServerEngines BladeEngine2 10Gb		
	NIC		
	c:\windows\system32\drivers\be2n		
d62.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
beep	Beep		
	c:\windows\system32\drivers\beep.		
sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
bfad	Brocade 425/825 4G/8G FC HBA		
	driver		
	c:\windows\system32\drivers\bfad.		
sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
bfad_up	Brocade 415/815 4G/8G FC HBA		
	filter driver		
	c:\windows\system32\drivers\bfad_		
up.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
blbdrive	blbdrive		
	c:\windows\system32\drivers\blbdri		
ve.sys	Kernel Driver Yes	System	
	Running OK	Normal	
	No Yes		
bnadimp	Brocade 10G Ethernet Service		
	c:\windows\system32\drivers\bnadi		
md6.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
bowser	Browser Support Driver		
	c:\windows\system32\drivers\bows		
er.sys	File System Driver	Yes	
	Manual Running	OK	
	Normal No	Yes	
brfiltlo	Brother USB Mass-Storage Lower		
	Filter Driver		
	c:\windows\system32\drivers\brfiltl		
o.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
brfiltup	Brother USB Mass-Storage Upper		
	Filter Driver		
	c:\windows\system32\drivers\brfiltu		
p.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
brserid	Brother MFC Serial Port Interface		
	Driver (WDM)		
	c:\windows\system32\drivers\brseri		
d.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
brserwdm	Brother WDM Serial driver		
	c:\windows\system32\drivers\brser		
wdm.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		

brusbmdm	Brother MFC USB Fax Only Modem		
	c:\windows\system32\drivers\brusb		
mdm.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
brusbser	Brother MFC USB Serial WDM Driver		
	c:\windows\system32\drivers\brusb		
ser.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
cdfs	CD/DVD File System Reader		
	c:\windows\system32\drivers\cdfs.s		
ys	File System Driver	No	
	Disabled Stopped	OK	
	Normal No	No	
cdrom	CD-ROM Driver		
	c:\windows\system32\drivers\cdro		
m.sys	Kernel Driver No	System	
	Stopped OK	Normal	
	No No		
clfs	Common Log (CLFS)		
	c:\windows\system32\clfs.sys		
	Kernel Driver Yes	Boot	
	Running OK	Critical	
	No Yes		
cmbatt	Microsoft ACPI Control Method		
	Battery Driver		
	c:\windows\system32\drivers\cmba		
tt.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
cmdide	cmdide		
	c:\windows\system32\drivers\cmdid		
e.sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
cng	CNG		
	c:\windows\system32\drivers\cng.s		
ys	Kernel Driver Yes	Boot	
	Running OK	Critical	
	No Yes		
compbatt	Compbatt		
	c:\windows\system32\drivers\comp		
batt.sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
compositebus	Composite Bus		
	Enumerator Driver		
	c:\windows\system32\drivers\comp		
ositebus.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
cpqteam	HP Network Configuration Utility		
	c:\windows\system32\drivers\cpqte		
am.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
crdisk	Crdisk Filter Driver		
	c:\windows\system32\drivers\crdis		
k.sys	Kernel Driver No	Disabled	
	Stopped OK	Normal	
	No No		
dfsc	DFS Namespace Client Driver		
	c:\windows\system32\drivers\dfsc.s		
ys	File System Driver	Yes	
	System Running	OK	
	Normal No	Yes	
discache	System Attribute Cache		
	c:\windows\system32\drivers\discac		
he.sys	Kernel Driver Yes	System	
	Running OK	Normal	
	No Yes		

disk	Disk Driver		
	c:\windows\system32\drivers\disk.s		
ys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
dxgkrnl	LDDM Graphics Subsystem		
	c:\windows\system32\drivers\dxgkr		
nl.sys	Kernel Driver No	Manual	
	Stopped OK	Ignore	
	No No		
e1express	Intel(R) PRO/1000 PCI Express		
	Network Connection Driver		
	c:\windows\system32\drivers\evbda		
032e.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
ebdrv	Broadcom NetXtreme II 10 GigE		
	VBD		
	c:\windows\system32\drivers\evbda		
.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
elxplus	Emulex PLUS Service		
	c:\windows\system32\drivers\elxplu		
s.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
elxstor	elxstor		
	c:\windows\system32\drivers\elxsto		
r.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
errdev	Microsoft Hardware Error Device		
	Driver		
	c:\windows\system32\drivers\errde		
v.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
exfat	exFAT File System Driver		
	c:\windows\system32\drivers\exfat.		
sys	File System Driver	No	
	Manual Stopped	OK	
	Normal No	No	
fastfat	FAT12/16/32 File System Driver		
	c:\windows\system32\drivers\fastfa		
t.sys	File System Driver	No	
	Manual Stopped	OK	
	Normal No	No	
fdc	Floppy Disk Controller Driver		
	c:\windows\system32\drivers\fdc.sy		
s	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
fileinfo	File Information FS MiniFilter		
	c:\windows\system32\drivers\fileinf		
o.sys	File System Driver	No	
	Manual Stopped	OK	
	Normal No	No	
filetrace	Filetrace		
	c:\windows\system32\drivers\filetra		
ce.sys	File System Driver	No	
	Manual Stopped	OK	
	Normal No	No	
flpydisk	Floppy Disk Driver		
	c:\windows\system32\drivers\flpydi		
sk.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		

fltMgr	FltMgr		
	c:\windows\system32\drivers\fltMgr		
.sys	File System Driver	Yes	
	Boot Running	OK	
	Critical No	Yes	
fsdepends	File System Dependency Minifilter		
	c:\windows\system32\drivers\fsdep		
ends.sys	File System Driver	No	
	Manual Stopped	OK	
	Critical No	No	
gagp30kx	Microsoft Generic AGPv3.0 Filter for		
	K8 Processor Platforms		
	c:\windows\system32\drivers\gagp		
30kx.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
hdaudbus	Microsoft UAA Bus Driver for High		
	Definition Audio		
	c:\windows\system32\drivers\hdau		
dbus.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
hidbatt	HID UPS Battery Driver		
	c:\windows\system32\drivers\hidba		
tt.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
hidusb	Microsoft HID Class Driver		
	c:\windows\system32\drivers\hidus		
b.sys	Kernel Driver Yes	Manual	
	Running OK	Ignore	
	No Yes		
hpciss2	HpCISSs2		
	c:\windows\system32\drivers\hpcis		
ss2.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
hpqilo3chif	HP ProLiant Management Controller		
	Driver (CHIF)		
	c:\windows\system32\drivers\hpqilo		
3chif.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
hpqilo3core	HP ProLiant Management Controller		
	Driver (CORE)		
	c:\windows\system32\drivers\hpqilo		
3core.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
hpqilo3whea	HP ProLiant Management Controller		
	Driver (WHEA)		
	c:\windows\system32\drivers\hpqilo		
3whea.sys	Kernel Driver Yes	System	
	Running OK	Normal	
	No Yes		
hpsamd	HpSAMD		
	c:\windows\system32\drivers\hpsa		
md.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		
http	HTTP		
	c:\windows\system32\drivers\http.s		
ys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
hwpolicy	Hardware Policy Driver		
	c:\windows\system32\drivers\hwpol		
icy.sys	Kernel Driver Yes	Boot	
	Running OK	Normal	
	No Yes		

i8042prt	i8042 Keyboard and PS/2 Mouse		
	Port Driver		
	c:\windows\system32\drivers\i8042		
prt.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
iastorv	IAStorV	No	
	c:\windows\system32\drivers\iastor		
v.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
ice	ice		
	c:\windows\system32\ice.sys		
	Kernel Driver Yes	Boot	
	Running OK	Ignore	
	No Yes		
iirsp	iirsp		
	c:\windows\system32\drivers\iirsp.s		
ys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
intelide	intelide		
	c:\windows\system32\drivers\intelid		
e.sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
intelppm	Intel Processor Driver		
	c:\windows\system32\drivers\intelp		
pm.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
ioatdma	Intel(R) QuickData Technology		
	Device		
	c:\windows\system32\drivers\qtd26		
0x64.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
ipfilterdriver	IP Traffic Filter Driver		
	c:\windows\system32\drivers\ipflttr		
v.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
ipmidrv	IPMIDRV		
	c:\windows\system32\drivers\ipmid		
rv.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
ipnat	IP Network Address Translator		
	c:\windows\system32\drivers\ipnat.		
sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
isapnp	isapnp		
	c:\windows\system32\drivers\isapn		
p.sys	Kernel Driver No	Manual	
	Stopped OK	Critical	
	No No		
iscsiprt	iScsiPort Driver		
	c:\windows\system32\drivers\msisc		
si.sys	Kernel Driver No	Manual	
	Stopped OK	Normal	
	No No		
kbdclass	Keyboard Class Driver		
	c:\windows\system32\drivers\kbddl		
ass.sys	Kernel Driver Yes	Manual	
	Running OK	Normal	
	No Yes		
kbdhid	Keyboard HID Driver		
	c:\windows\system32\drivers\kbddi		
d.sys	Kernel Driver Yes	Manual	
	Running OK	Ignore	
	No Yes		

kldbgdrv	kldbgdrv \\??\c:\windows\system32\kldbgdrv.		
sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
ksecdd	KSecDD		
d.sys	c:\windows\system32\drivers\ksecdd	Kernel Driver	Yes
	Running	OK	Critical
	No	Yes	
ksecpkg	KSecPkg		
kg.sys	c:\windows\system32\drivers\ksecp	Kernel Driver	Yes
	Running	OK	Critical
	No	Yes	
ksthunk	Kernel Streaming Thunks		
nk.sys	c:\windows\system32\drivers\ksth	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
l2nd	Broadcom NetXtreme II BXND		
60a.sys	c:\windows\system32\drivers\bxnd	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
ltdio	Link-Layer Topology Discovery		
	Mapper I/O Driver		
sys	c:\windows\system32\drivers\ltdio.	Kernel Driver	Yes
	Running	OK	Normal
	No	Yes	
lsi_fc	LSI_FC		
sys	c:\windows\system32\drivers\lsi_fc.	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
lsi_sas	LSI_SAS		
s.sys	c:\windows\system32\drivers\lsi_sa	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
lsi_sas2	LSI_SAS2		
s2.sys	c:\windows\system32\drivers\lsi_sa	Kernel Driver	Yes
	Running	OK	Normal
	No	Yes	
lsi_scsi	LSI_SCSI		
si.sys	c:\windows\system32\drivers\lsi_sc	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
luafr	UAC File Virtualization		
sys	c:\windows\system32\drivers\luafr.	File System Driver	Yes
	Auto	Running	OK
	Normal	No	Yes
megagas	megagas		
sys	c:\windows\system32\drivers\mega	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
megasr	MegaSR		
sr.sys	c:\windows\system32\drivers\mega	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
modem	Modem		
m.sys	c:\windows\system32\drivers\mode	Kernel Driver	No
	Stopped	OK	Ignore
	No	No	

monitor	Microsoft Monitor Class Function		
	Driver Service		
or.sys	c:\windows\system32\drivers\monit	Kernel Driver	Yes
	Running	OK	Manual
	No	Normal	
mouclass	Mouse Class Driver		
ass.sys	c:\windows\system32\drivers\moucl	Kernel Driver	Yes
	Running	OK	Normal
	No	Yes	
mouhid	Mouse HID Driver		
id.sys	c:\windows\system32\drivers\mouh	Kernel Driver	Yes
	Running	OK	Ignore
	No	Yes	
mountmgr	Mount Point Manager		
tmgr.sys	c:\windows\system32\drivers\moun	Kernel Driver	Yes
	Running	OK	Critical
	No	Yes	
mpio	Microsoft Multi-Path Bus Driver		
sys	c:\windows\system32\drivers\mpio.	Kernel Driver	Yes
	Running	OK	Normal
	No	Yes	
mpsdrv	Windows Firewall Authorization		
	Driver		
rv.sys	c:\windows\system32\drivers\mpsd	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
mrxsm	SMB MiniRedirector Wrapper and		
	Engine		
mb.sys	c:\windows\system32\drivers\mrxs	File System Driver	Yes
	Manual	Running	OK
	Normal	No	Yes
mrxsm10	SMB 1.x MiniRedirector		
mb10.sys	c:\windows\system32\drivers\mrxs	File System Driver	Yes
	Manual	Running	OK
	Normal	No	Yes
mrxsm20	SMB 2.0 MiniRedirector		
mb20.sys	c:\windows\system32\drivers\mrxs	File System Driver	Yes
	Manual	Running	OK
	Normal	No	Yes
msahci	msahci		
ci.sys	c:\windows\system32\drivers\msah	Kernel Driver	No
	Stopped	OK	Critical
	No	No	
msdsm	Microsoft Multi-Path Device Specific		
	Module		
m.sys	c:\windows\system32\drivers\msds	Kernel Driver	Yes
	Running	OK	Normal
	No	Yes	
msfs	Msfs		
sys	c:\windows\system32\drivers\msfs.	File System Driver	Yes
	System	Running	OK
	Normal	No	Yes
mshidkmdf	Pass-through HID to KMDf Filter		
	Driver		
kmdf.sys	c:\windows\system32\drivers\mshid	Kernel Driver	No
	Stopped	OK	Ignore
	No	No	

msisadvr	msisadvr		
drv.sys	c:\windows\system32\drivers\msisa	Kernel Driver	Yes
	Running	OK	Critical
msrpc	MsRPC		
.sys	c:\windows\system32\drivers\msrpc	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
mssmbios	Microsoft System Management		
	BIOS Driver		
bios.sys	c:\windows\system32\drivers\mssm	Kernel Driver	Yes
	Running	OK	System
	No	Yes	Normal
mtconfig	Microsoft Input Configuration Driver		
nfig.sys	c:\windows\system32\drivers\mtco	Kernel Driver	No
	Stopped	OK	Normal
	No	No	
mup	Mup		
sys	c:\windows\system32\drivers\mup.	File System Driver	Yes
	Boot	Running	OK
	Normal	No	Yes
ndis	NDIS System Driver		
ys	c:\windows\system32\drivers\ndis.s	Kernel Driver	Yes
	Running	OK	Boot
	No	Yes	Critical
ndiscap	NDIS Capture LightWeight Filter		
ap.sys	c:\windows\system32\drivers\ndisc	Kernel Driver	No
	Stopped	OK	Manual
	No	No	Normal
ndistapi	Remote Access NDIS TAPI Driver		
pi.sys	c:\windows\system32\drivers\ndista	Kernel Driver	Yes
	Running	OK	Manual
	No	Yes	Normal
ndisui	NDIS Usermode I/O Protocol		
o.sys	c:\windows\system32\drivers\ndisui	Kernel Driver	No
	Stopped	OK	Manual
	No	No	Normal
ndiswan	Remote Access NDIS WAN Driver		
an.sys	c:\windows\system32\drivers\ndisw	Kernel Driver	Yes
	Running	OK	Manual
	No	Yes	Normal
ndproxy	NDIS Proxy		
xy.sys	c:\windows\system32\drivers\ndpro	Kernel Driver	Yes
	Running	OK	Manual
	No	Yes	Normal
netbios	NetBIOS Interface		
os.sys	c:\windows\system32\drivers\netbi	File System Driver	Yes
	System	Running	OK
	Normal	No	Yes
netbt	NetBT		
sys	c:\windows\system32\drivers\netbt.	Kernel Driver	Yes
	Running	OK	System
	No	Yes	Normal
nfrd960	nfrd960		
60.sys	c:\windows\system32\drivers\nfrd9	Kernel Driver	No
	Stopped	OK	Manual
	No	No	Normal

npfs	Npfs		
	c:\windows\system32\drivers\npfs.s		
ys	File System Driver	Yes	
	System	Running	OK
	Normal	No	Yes
nsiproxy	NSI proxy service driver.		
	c:\windows\system32\drivers\nsipr		
oxy.sys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
ntfs	Ntfs		
	c:\windows\system32\drivers\ntfs.s		
ys	File System Driver	Yes	
	Manual	Running	OK
	Normal	No	Yes
null	Null		
	c:\windows\system32\drivers\null.s		
ys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
nvraid	nvraid		
	c:\windows\system32\drivers\nvrai		
d.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
nvstor	nvstor		
	c:\windows\system32\drivers\nvsto		
r.sys	Kernel Driver	No	Manual
	Stopped	OK	Critical
	No	No	
nv_agp	NVIDIA nForce AGP Bus Filter		
	c:\windows\system32\drivers\nv_ag		
p.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
ohci1394	1394 OHCI Compliant Host		
Controller (Legacy)			
	c:\windows\system32\drivers\ohci1		
394.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
parport	Parallel port driver		
	c:\windows\system32\drivers\parpo		
rt.sys	Kernel Driver	No	Manual
	Stopped	OK	Ignore
	No	No	
partmgr	Partition Manager		
	c:\windows\system32\drivers\partm		
gr.sys	Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
pci	PCI Bus Driver		
	c:\windows\system32\drivers\pci.sy		
s	Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
pciide	pciide		
	c:\windows\system32\drivers\pciide		
.sys	Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
pcmcia	pcmcia		
	c:\windows\system32\drivers\pcmc		
a.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
pcw	Performance Counters for Windows		
Driver			
	c:\windows\system32\drivers\pcw.s		
ys	Kernel Driver	Yes	Boot
	Running	OK	Normal
	No	Yes	

peauth	PEAUTH		
	c:\windows\system32\drivers\peaut		
h.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
pptpminiport	WAN Miniport (PPTP)		
	c:\windows\system32\drivers\raspp		
tp.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
processor	Processor Driver		
	c:\windows\system32\drivers\proce		
ssr.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
psched	QoS Packet Scheduler		
	c:\windows\system32\drivers\pacer		
.sys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
ql2300	QLogic Fibre Channel STOR		
Miniport Driver (wx64 IP)			
	c:\windows\system32\drivers\ql230		
0.sys	Kernel Driver	Yes	Boot
	Running	OK	Normal
	No	Yes	
ql40xx	ql40xx		
	c:\windows\system32\drivers\ql40x		
x.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
rasacd	Remote Access Auto Connection		
Driver			
	c:\windows\system32\drivers\rasac		
d.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
rasagilevpn	WAN Miniport (IKEv2)		
	c:\windows\system32\drivers\agilev		
pn.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasl2tp	WAN Miniport (L2TP)		
	c:\windows\system32\drivers\rasl2t		
p.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasppoe	Remote Access PPPOE Driver		
	c:\windows\system32\drivers\raspp		
poe.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasstp	WAN Miniport (SSTP)		
	c:\windows\system32\drivers\rasstp		
p.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rdbss	Redirected Buffering Sub System		
	c:\windows\system32\drivers\rdbss.		
sys	File System Driver	Yes	
	System	Running	OK
	Normal	No	Yes
rdpbus	Remote Desktop Device Redirector		
Bus Driver			
	c:\windows\system32\drivers\rdpbu		
s.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rdpcdd	RDPCDD		
	c:\windows\system32\drivers\rdpcd		
d.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	

rdpdr	Terminal Server Device Redirector		
Driver			
	c:\windows\system32\drivers\rdpdr.		
sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
rdpencdd	RDP Encoder Driver		
	c:\windows\system32\drivers\rdpen		
cdd.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
rdpprefmp	Reflector Display Driver used to		
gain access to graphics data			
	c:\windows\system32\drivers\rdpref		
mp.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
rdpwd	RDP Winstation Driver		
	c:\windows\system32\drivers\rdpw		
d.sys	Kernel Driver	Yes	Manual
	Running	OK	Ignore
	No	Yes	
rspndr	Link-Layer Topology Discovery		
Responder			
	c:\windows\system32\drivers\rspnd		
r.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
	No	Yes	
s3cap	s3cap		
	c:\windows\system32\drivers\vms3		
cap.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
sacdrv	sacdrv		
	c:\windows\system32\drivers\sacdr		
v.sys	Kernel Driver	No	Boot
	Stopped	OK	Ignore
	No	No	
sbp2port	sbp2port		
	c:\windows\system32\drivers\sbp2p		
ort.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
scfilter	Smart card PnP Class Filter Driver		
	c:\windows\system32\drivers\scfite		
r.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
secdrv	Security Driver		
	c:\windows\system32\drivers\secdr		
v.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
	No	Yes	
serenum	Serenum Filter Driver		
	c:\windows\system32\drivers\seren		
um.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
serial	Serial port driver		
	c:\windows\system32\drivers\serial.		
sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
sermouse	Serial Mouse Driver		
	c:\windows\system32\drivers\serm		
ouse.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
sfddisk	SFF Storage Class Driver		
	c:\windows\system32\drivers\sfddis		
k.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	

sffp_mmc	SFF Storage Protocol Driver for MMC		
mmc.sys	c:\windows\system32\drivers\sffp_	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
sffp_sd	SFF Storage Protocol Driver for SDBus		
d.sys	c:\windows\system32\drivers\sffp_s	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
sfloppy	High-Capacity Floppy Disk Drive		
y.sys	c:\windows\system32\drivers\sflopp	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
sisraid2	SiSRaid2		
d2.sys	c:\windows\system32\drivers\sisrai	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
sisraid4	SiSRaid4		
d4.sys	c:\windows\system32\drivers\sisrai	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
smb	Message-oriented TCP/IP and TCP/IPV6 Protocol (SMB session)		
ys	c:\windows\system32\drivers\smb.s	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
spldr	Security Processor Loader Driver		
sys	c:\windows\system32\drivers\spldr.	Kernel Driver Yes	Boot
		Running OK	Critical
		No	Yes
srv	Server SMB 1.xxx Driver		
s	c:\windows\system32\drivers\srv.sy	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
srv2	Server SMB 2.xxx Driver		
ys	c:\windows\system32\drivers\srv2.s	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
srvnet	srvnet		
t.sys	c:\windows\system32\drivers\srve	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
stexstor	stexstor		
or.sys	c:\windows\system32\drivers\stexst	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
storflt	Disk Virtual Machine Bus Acceleration Filter Driver		
rfl.sys	c:\windows\system32\drivers\vmsto	Kernel Driver Yes	Boot
		Running OK	Normal
		No	Yes
storvsc	storvsc		
c.sys	c:\windows\system32\drivers\storvs	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No

storvsp	storvsp		
p.sys	c:\windows\system32\drivers\storvs	Kernel Driver No	Manual
		Stopped OK	Normal
swenum	Software Bus Enumerator		
um.sys	c:\windows\system32\drivers\swen	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
tcpip	TCP/IP Protocol Driver		
sys	c:\windows\system32\drivers\tcpip.	Kernel Driver Yes	Boot
		Running OK	Normal
		No	Yes
tcpip6	Microsoft IPv6 Protocol Driver		
sys	c:\windows\system32\drivers\tcpip.	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
tcpipreg	TCP/IP Registry Compatibility		
eg.sys	c:\windows\system32\drivers\tcpipr	Kernel Driver Yes	Auto
		Running OK	Normal
		No	Yes
tdpipe	TDPIPE		
e.sys	c:\windows\system32\drivers\tdpip	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
tdtcp	TDTCP		
sys	c:\windows\system32\drivers\tdtcp.	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
tdx	NetIO Legacy TDI Support Driver		
s	c:\windows\system32\drivers\tdx.sy	Kernel Driver Yes	System
		Running OK	Normal
		No	Yes
termdd	Terminal Device Driver		
d.sys	c:\windows\system32\drivers\termd	Kernel Driver Yes	System
		Running OK	Normal
		No	Yes
tssecsrv	Remote Desktop Services Security Filter Driver		
rv.sys	c:\windows\system32\drivers\tssecs	Kernel Driver Yes	Manual
		Running OK	Ignore
		No	Yes
tunnel	Microsoft Tunnel Miniport Adapter Driver		
l.sys	c:\windows\system32\drivers\tunne	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
uagp35	Microsoft AGPv3.5 Filter		
35.sys	c:\windows\system32\drivers\uagp	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
udfs	udfs		
ys	c:\windows\system32\drivers\udfs.s	File System Driver	No
		Disabled Stopped	OK
		Normal No	No
uliagpkx	Uli AGP Bus Filter		
kx.sys	c:\windows\system32\drivers\uliagp	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No

umbus	UMBus Enumerator Driver		
s.sys	c:\windows\system32\drivers\umbu	Kernel Driver Yes	Manual
		Running OK	Normal
umpass	Microsoft UMPass Driver		
ss.sys	c:\windows\system32\drivers\umpa	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
usbccgp	Microsoft USB Generic Parent Driver		
gp.sys	c:\windows\system32\drivers\usbcc	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
usbhci	Microsoft USB 2.0 Enhanced Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usbh	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
usbhub	Microsoft USB Standard Hub Driver		
b.sys	c:\windows\system32\drivers\usbhu	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
usbohci	Microsoft USB Open Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usboh	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
usbprint	Microsoft USB PRINTER Class		
int.sys	c:\windows\system32\drivers\usbpr	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
usbstor	USB Mass Storage Driver		
or.sys	c:\windows\system32\drivers\usbst	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
usbuhci	Microsoft USB Universal Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usbuh	Kernel Driver Yes	Manual
		Running OK	Normal
		No	Yes
vdroot	Microsoft Virtual Drive Enumerator Driver		
oot.sys	c:\windows\system32\drivers\vdrr	Kernel Driver Yes	Boot
		Running OK	Critical
		No	Yes
vga	vga		
p.sys	c:\windows\system32\drivers\vgapn	Kernel Driver Yes	Manual
		Running OK	Ignore
		No	Yes
vgasave	VgaSave		
ys	c:\windows\system32\drivers\vgas	Kernel Driver Yes	System
		Running OK	Ignore
		No	Yes
vhdmp	vhdmp		
p.sys	c:\windows\system32\drivers\vhdmp	Kernel Driver No	Manual
		Stopped OK	Normal
		No	No
viaide	viaide		
.sys	c:\windows\system32\drivers\viaide	Kernel Driver No	Manual
		Stopped OK	Critical
		No	No

```

vid Vid
c:\windows\system32\drivers\vid.sys
s Kernel Driver No Manual
Stopped OK Normal
No No
vmbus Virtual Machine Bus
c:\windows\system32\drivers\vmbru
s.sys Kernel Driver No Manual
Stopped OK Normal
No No
vmbushid VMBusHID
c:\windows\system32\drivers\vmbru
shid.sys Kernel Driver No Manual
Stopped OK Ignore
No No
volmgr Volume Manager Driver
c:\windows\system32\drivers\volmgr
r.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
volmgrx Dynamic Volume Manager
c:\windows\system32\drivers\volmgr
rx.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
volsnap Storage volumes
c:\windows\system32\drivers\volsn
ap.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
vsmraid vsmraid
c:\windows\system32\drivers\vsmra
id.sys Kernel Driver No Manual
Stopped OK Normal
No No
wacompen Wacom Serial Pen HID Driver
c:\windows\system32\drivers\waco
mpen.sys Kernel Driver No Manual
Stopped OK Normal
No No
wanarp Remote Access IP ARP Driver
c:\windows\system32\drivers\wana
rp.sys Kernel Driver No Manual
Stopped OK Normal
No No
wanarpv6 Remote Access IPv6 ARP Driver
c:\windows\system32\drivers\wana
rp.sys Kernel Driver Yes System
Running OK Normal
No Yes
wd Wd
c:\windows\system32\drivers\wd.sys
s Kernel Driver No Manual
Stopped OK Normal
No No
wdf01000 Kernel Mode Driver Frameworks
service c:\windows\system32\drivers\wdf0
1000.sys Kernel Driver Yes Boot
Running OK Normal
No Yes
wfpwfp WFP Lightweight Filter
c:\windows\system32\drivers\wfpwfp
f.sys Kernel Driver Yes System
Running OK Normal
No Yes
wimmount WIMMount
c:\windows\system32\drivers\wimm
ount.sys File System Driver No
Manual Stopped OK
Normal No No

```

```

wmiacpi Microsoft Windows Management
Interface for ACPI
c:\windows\system32\drivers\wmiacpi.sys
Kernel Driver No Manual
Stopped OK Normal
ws2ifsl Winsock IFS Driver
c:\windows\system32\drivers\ws2ifsl.sys
Kernel Driver No Disabled
Stopped OK Normal
No No
wudfpf User Mode Driver Frameworks
Platform Driver
c:\windows\system32\drivers\wudfpf.sys
Kernel Driver No Manual
Stopped OK Normal
No No

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe
<SYSTEM>
FP_NO_HOST_CHECK NO
<SYSTEM>
OS Windows_NT <SYSTEM>
Path C:\Program Files
(x86)\BROCADE\Adapter\driver;C:\Program
Files\HP\NCU\C:\Program Files
(x86)\Brocade\DriverPackage\bfa\util;%System
Root%\system32;%SystemRoot%;%SystemRoot
%\System32\Wbem;%SYSTEMROOT%\System3
2\WindowsPowerShell\v1.0;C:\Program Files
(x86)\Microsoft SQL
Server\80\Tools\Binn\C:\Program Files\Microsoft
SQL Server\90\Tools\Binn\C:\Program Files
(x86)\Microsoft SQL
Server\90\Tools\Binn\C:\Program Files\Microsoft
SQL Server\90\Tools\Binn\ <SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.J
S;.JSE;.WSF;.WSH;.MSC <SYSTEM>
PROCESSOR_ARCHITECTURE AMD64
<SYSTEM>
TEMP %SystemRoot%\TEMP
<SYSTEM>
TMP %SystemRoot%\TEMP
<SYSTEM>
USERNAME SYSTEM <SYSTEM>
windir %SystemRoot%
<SYSTEM>
PSModulePath
%SystemRoot%\system32\Window
sPowerShell\v1.0\Modules\ <SYSTEM>
NUMBER_OF_PROCESSORS 48
<SYSTEM>
PROCESSOR_LEVEL 16
<SYSTEM>
PROCESSOR_IDENTIFIER AMD64 Family 16
Model 9 Stepping 1, AuthenticAMD
<SYSTEM>
PROCESSOR_REVISION 0901
<SYSTEM>
TEMP
emp %USERPROFILE%\AppData\Local\T
TMP NT AUTHORITY\SYSTEM
emp %USERPROFILE%\AppData\Local\T
TMP NT AUTHORITY\SYSTEM
emp %USERPROFILE%\AppData\Local\T
TMP NT AUTHORITY\LOCAL SERVICE

```

```

TMP %USERPROFILE%\AppData\Local\T
emp NT AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\AppData\Local\T
emp SERVICE
TMP NT AUTHORITY\NETWORK
emp SERVICE
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\guz
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\guz
Path C:\Program Files
(x86)\BROCADE\Adapter\driver;C:\Program
Files\HP\NCU\C:\Program Files
(x86)\Brocade\DriverPackage\bfa\util;%System
Root%\system32;%SystemRoot%;%SystemRoot
%\System32\Wbem;%SYSTEMROOT%\System3
2\WindowsPowerShell\v1.0;C:\Program Files
(x86)\Microsoft SQL
Server\80\Tools\Binn\C:\Program Files\Microsoft
SQL Server\90\Tools\Binn\C:\Program Files
(x86)\Microsoft SQL
Server\90\Tools\Binn\C:\Program Files\Microsoft
SQL Server\90\Tools\Binn\ SQLWESTMERE\guz
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\Admin2
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\Admin2
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\Administrator
TMP %USERPROFILE%\AppData\Local\T
emp SQLWESTMERE\Administrator
PATH C:\Program Files
(x86)\BROCADE\Adapter\driver
SQLWESTMERE\Administrator

[Print Jobs]

Can't Collect Information

[Network Connections]

Local Name Remote Name Type
Status User Name
X: \\symbols\files Disk
Persistent Connection
Y: \\dbg\privates Disk
Persistent Connection
Z: \\symbols\symbols Disk
Persistent Connection

[Running Tasks]

Name Path Process ID Priority
Working Set Min Working Set Max
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
                        Not Available Not Available
                        1/7/2011 1:40 PM Not
Available Not Available Not Available
smss.exe Not Available 592      11
                        200      1380
                        1/7/2011 1:40 PM Not
Available Not Available Not Available
csrss.exe c:\windows\system32\csrss.exe
                        764      13      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
wininit.exe c:\windows\system32\wininit.exe
                        804      13      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      126.00
KB (129,024 bytes) 7/13/2009 4:52 PM

csrss.exe c:\windows\system32\csrss.exe
                        816      13      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
services.exe c:\windows\system32\services.exe
                        868      9      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      321.00
KB (328,704 bytes) 7/13/2009 4:19 PM

lsass.exe c:\windows\system32\lsass.exe
                        876      9      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      30.50 KB
(31,232 bytes) 7/13/2009 4:20 PM

lsm.exe c:\windows\system32\lsm.exe
                        884      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      325.50
KB (333,312 bytes) 7/13/2009 5:17 PM

winlogon.exe c:\windows\system32\winlogon.exe
                        932      13      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16447      380.50
KB (389,632 bytes) 3/3/2010 4:45 PM

svchost.exe c:\windows\system32\svchost.exe
                        640      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        708      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        744      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

```

```

logonui.exe c:\windows\system32\logonui.exe
                        1028      13      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      27.00 KB
(27,648 bytes) 7/13/2009 4:52 PM
svchost.exe c:\windows\system32\svchost.exe
                        1080      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        1136      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        1184      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        1224      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        1392      8      200
                        1380      1/7/2011 1:40 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        2472      8      200
                        1380      1/7/2011 1:41 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
                        2960      8      200
                        1380      1/7/2011 1:43 PM
                        6.1.7600.16385      26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

msdtc.exe c:\windows\system32\msdtc.exe
                        3020      8      200
                        1380      1/7/2011 1:43 PM
                        2001.12.8530.16385      138.50
KB (141,824 bytes) 7/13/2009 4:59 PM

csrss.exe c:\windows\system32\csrss.exe
                        1936      13      200
                        1380      1/7/2011 1:46 PM
                        6.1.7600.16385      7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
winlogon.exe c:\windows\system32\winlogon.exe
                        1048      13      200
                        1380      1/7/2011 1:46 PM
                        6.1.7600.16447      380.50
KB (389,632 bytes) 3/3/2010 4:45 PM

taskhost.exe c:\windows\system32\taskhost.exe
                        2600      8      200
                        1380      1/7/2011 1:51 PM
                        6.1.7600.16385      67.50 KB
(69,120 bytes) 7/13/2009 4:31 PM

```

```

rdpclip.exe c:\windows\system32\rdpclip.exe
                        360      8      200
                        1380      1/7/2011 1:51 PM
                        6.1.7600.16385      204.50
KB (209,408 bytes) 7/13/2009 5:17 PM
dwm.exe c:\windows\system32\dwm.exe
                        2696      8      200
                        1380      1/7/2011 1:51 PM
                        6.1.7600.16385      117.50
KB (120,320 bytes) 7/13/2009 4:37 PM

explorer.exe c:\windows\explorer.exe 324
                        8      200      1380
                        1/7/2011 1:51 PM
                        6.1.7600.16450      2.74 MB
(2,870,272 bytes) 3/3/2010 4:45 PM

taskmgr.exe c:\windows\system32\taskmgr.exe
                        912      13      200
                        1380      1/9/2011 11:21 PM
                        6.1.7600.16385      251.00
KB (257,024 bytes) 7/13/2009 4:31 PM

cmd.exe c:\windows\system32\cmd.exe
                        2000      8      200
                        1380      1/13/2011 12:51 PM
                        6.1.7600.16385      336.50
KB (344,576 bytes) 7/13/2009 4:34 PM

conhost.exe c:\windows\system32\conhost.exe
                        3616      8      200
                        1380      1/13/2011 12:51 PM
                        6.1.7600.16385      330.50
KB (338,432 bytes) 7/13/2009 4:38 PM

sqlservr.exe c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe 2872
                        13      200      1380
                        1/13/2011 12:51 PM
                        2005.90.4267.0      36.94
MB (38,734,168 bytes) 10/16/2009 11:23 PM

msinfo32.exe c:\windows\system32\msinfo32.exe
                        3152      8      200
                        1380      1/13/2011 3:27 PM
                        6.1.7600.16385      370.00
KB (378,880 bytes) 7/13/2009 4:31 PM

wmiprvse.exe
c:\windows\system32\wbem\wmiprvse.exe
                        3116      8      200
                        1380      1/13/2011 3:27 PM
                        6.1.7600.16385      360.00
KB (368,640 bytes) 7/13/2009 4:47 PM

wmiprvse.exe
c:\windows\system32\wbem\wmiprvse.exe
                        3220      8      200
                        1380      1/13/2011 3:33 PM
                        6.1.7600.16385      360.00
KB (368,640 bytes) 7/13/2009 4:47 PM

[Loaded Modules]

Name Version Size File Date
ManufacturerPath
csrss 6.1.7600.16385 7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\csrss.exe

```

ntdll 6.1.7600.16385 1.66 MB
(1,736,792 bytes) 7/13/2009 4:22 PM
Microsoft Corporation
c:\windows\system32\ntdll.dll

csrssrv 6.1.7600.16385 42.50 KB
(43,520 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\csrssrv.dll

basesrv 6.1.7600.16385 51.50 KB
(52,736 bytes) 7/13/2009 4:18 PM
Microsoft Corporation
c:\windows\system32\basesrv.dll

winsrv 6.1.7600.16385 209.00
KB (214,016 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\winsrv.dll

user32 6.1.7600.16385 985.00
KB (1,008,640 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\user32.dll

gdi32 6.1.7600.16385 395.00
KB (404,480 bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\gdi32.dll

kernel32 6.1.7600.16385 1.11 MB
(1,162,240 bytes) 7/13/2009 4:28 PM
Microsoft Corporation
c:\windows\system32\kernel32.dll

kernelbase 6.1.7600.16385 411.50
KB (421,376 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\kernelbase.dll

lpk 6.1.7600.16385 41.00 KB
(41,984 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\lpk.dll

usp10 1.626.7600.16385 782.50
KB (801,280 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\usp10.dll

msvcrt 7.0.7600.16385 620.00
KB (634,880 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\msvcrt.dll

sxssrv 6.1.7600.16385 31.00 KB
(31,744 bytes) 7/13/2009 4:26 PM
Microsoft Corporation
c:\windows\system32\sxssrv.dll

sxs 6.1.7600.16385 569.50
KB (583,168 bytes) 7/13/2009 4:27 PM
Microsoft Corporation
c:\windows\system32\sxs.dll

rpcrt4 6.1.7600.16385 1.17 MB
(1,221,632 bytes) 7/13/2009 4:23 PM
Microsoft Corporation
c:\windows\system32\rpcrt4.dll

cryptbase 6.1.7600.16385 43.00 KB
(44,032 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\cryptbase.dll

advapi32 6.1.7600.16385 856.50
KB (877,056 bytes) 7/13/2009 5:41 PM
Microsoft Corporation
c:\windows\system32\advapi32.dll

sechost 6.1.7600.16385 111.00
KB (113,664 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\sechost.dll

wininit 6.1.7600.16385 126.00
KB (129,024 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\wininit.exe

profapi 6.1.7600.16385 43.00 KB
(44,032 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\profapi.dll

imm32 6.1.7600.16385 163.50
KB (167,424 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\imm32.dll

msctf 6.1.7600.16385 1.02 MB
(1,067,008 bytes) 7/13/2009 4:40 PM
Microsoft Corporation
c:\windows\system32\msctf.dll

rpctrremote 6.1.7600.16385 63.50 KB
(65,024 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\rpctrremote.dll

apphelp 6.1.7600.16385 330.50
KB (338,432 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\apphelp.dll

ws2_32 6.1.7600.16385 289.50
KB (296,448 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\ws2_32.dll

nsi 6.1.7600.16385 13.50 KB
(13,824 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\nsi.dll

msocket 6.1.7600.16385 312.50
KB (320,000 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\msocket.dll

wshtcpip 6.1.7600.16385 13.00 KB
(13,312 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\wshtcpip.dll

wship6 6.1.7600.16385 13.50 KB
(13,824 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\wship6.dll

secur32 6.1.7600.16385 27.50 KB
(28,160 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\secur32.dll

sspicli 6.1.7600.16385 133.00
KB (136,192 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\sspicli.dll

credssp 6.1.7600.16385 20.00 KB
(20,480 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\credssp.dll

services 6.1.7600.16385 321.00
KB (328,704 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\services.exe

scext 6.1.7600.16385 87.00 KB
(89,088 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\scext.dll

scsvr 6.1.7600.16385 396.50
KB (406,016 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\scsvr.dll

svrcli 6.1.7600.16385 124.50
KB (127,488 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\svrcli.dll

authz 6.1.7600.16385 173.50
KB (177,664 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\authz.dll

ubpm 6.1.7600.16385 209.00
KB (214,016 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\ubpm.dll

wtsapi32 6.1.7600.16385 53.00 KB
(54,272 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\wtsapi32.dll

winsta 6.1.7600.16385 228.00
KB (233,472 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\winsta.dll

lsass 6.1.7600.16385 30.50 KB
(31,232 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\lsass.exe

sspsrv 6.1.7600.16385 28.00 KB
(28,672 bytes) 7/13/2009 4:20 PM
Microsoft Corporation
c:\windows\system32\sspsrv.dll

lsasrv 6.1.7600.16484 1.38 MB
(1,446,912 bytes) 10/5/2010 8:37 AM
Microsoft Corporation
c:\windows\system32\lsasrv.dll

samsrv 6.1.7600.16385 740.00
KB (757,760 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\samsrv.dll

cryptdll 6.1.7600.16385 64.50 KB
(66,048 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cryptdll.dll

msasn1 6.1.7600.16415 45.50 KB
(46,592 bytes) 3/3/2010 4:45 PM
Microsoft Corporation
c:\windows\system32\msasn1.dll

wevtapi 6.1.7600.16385 418.00
KB (428,032 bytes) 7/13/2009 4:46 PM
Microsoft Corporation
c:\windows\system32\wevtapi.dll

cngaudit 6.1.7600.16385 18.50 KB
(18,944 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cngaudit.dll

ncrypt 6.1.7600.16385 300.00
KB (307,200 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\ncrypt.dll

bcrypt 6.1.7600.16385 121.00
KB (123,904 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\bcrypt.dll

msprivs 6.1.7600.16385 2.00 KB
(2,048 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\msprivs.dll

netjoin 6.1.7600.16385 184.50
KB (188,928 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netjoin.dll

negoexts 6.1.7600.16385 114.50
KB (117,248 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\negoexts.dll

kerberos 6.1.7600.16385 697.50
KB (714,240 bytes) 7/13/2009 4:51 PM
Microsoft Corporation
c:\windows\system32\kerberos.dll

cryptsp 6.1.7600.16385 78.00 KB
(79,872 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\cryptsp.dll

msv1_0 6.1.7600.16420 304.50
KB (311,808 bytes) 3/3/2010 4:45 PM
Microsoft Corporation
c:\windows\system32\msv1_0.dll

netlogon 6.1.7600.16385 676.50
KB (692,736 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netlogon.dll

dnsapi 6.1.7600.16385 348.00
KB (356,352 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dnsapi.dll

logoncli 6.1.7600.16385 182.00
KB (186,368 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\logoncli.dll

schannel 6.1.7600.16612 333.00
KB (340,992 bytes) 10/5/2010 8:37 AM
Microsoft Corporation
c:\windows\system32\schannel.dll

crypt32 6.1.7600.16385 1.39 MB
(1,454,592 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\crypt32.dll

wdigest 6.1.7600.16385 205.50
KB (210,432 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\wdigest.dll

rsaenh 6.1.7600.16385 274.66
KB (281,256 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\rsaenh.dll

tspkg 6.1.7600.16385 84.00 KB
(86,016 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\tspkg.dll

pku2u 6.1.7600.16385 235.00
KB (240,640 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\pku2u.dll

bcryptprimitives 6.1.7600.16385
291.32 KB (298,312 bytes)
7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\bcryptprimitiv
es.dll

eflsaext 6.1.7600.16385 55.50 KB
(56,832 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\eflsaext.dll

scecli 6.1.7600.16385 227.00
KB (232,448 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\scecli.dll

rassfm 6.1.7600.16385 28.50 KB
(29,184 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\rassfm.dll

efssvc 6.1.7600.16385 36.50 KB
(37,376 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\efssvc.dll

efscore 6.1.7600.16385 297.00
KB (304,128 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\efscore.dll

efsutil 6.1.7600.16385 34.00 KB
(34,816 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\efsutil.dll

userenv 6.1.7600.16385 104.50
KB (107,008 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\userenv.dll

slc 6.1.7600.16385 30.00 KB
(30,720 bytes) 7/13/2009 4:51 PM
Microsoft Corporation
c:\windows\system32\slc.dll

gpapi 6.1.7600.16385 94.50 KB
(96,768 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\gpapi.dll

dssenh 6.1.7600.16385 186.41
KB (190,880 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\dssenh.dll

cryptnet 6.1.7600.16385 135.50
KB (138,752 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cryptnet.dll

wldap32 6.1.7600.16385 304.50
KB (311,808 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\wldap32.dll

iphlpapi 6.1.7600.16385 142.50
KB (145,920 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\iphlpapi.dll

winsnsi 6.1.7600.16385 25.50 KB
(26,112 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\winsnsi.dll

netutils 6.1.7600.16385 28.00 KB
(28,672 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netutils.dll

samcli 6.1.7600.16385 65.50 KB
(67,072 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\samcli.dll

samlib 6.1.7600.16385 104.50
KB (107,008 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\samlib.dll

certpoleng 6.1.7600.16385 70.00 KB
(71,680 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\certpoleng.dll

lsm 6.1.7600.16385 325.50
KB (333,312 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\lsm.exe

sysntfy 6.1.7600.16385 22.50 KB
(23,040 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\sysntfy.dll

wmsgapi 6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\wmsgapi.dll

pcwum 6.1.7600.16385 36.00 KB
(36,864 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\pcwum.dll

ole32 6.1.7600.16385 1.99 MB
(2,084,352 bytes) 7/13/2009 5:02 PM
Microsoft Corporation
c:\windows\system32\ole32.dll

ntmarta 6.1.7600.16385 158.50
KB (162,304 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\ntmarta.dll

clbcatq 2001.12.8530.16385 593.50
KB (607,744 bytes) 7/13/2009 5:00 PM
Microsoft Corporation
c:\windows\system32\clbcatq.dll

oleaut32	6.1.7600.16385	841.00 KB (861,184 bytes)	7/13/2009 4:59 PM	Microsoft Corporation	c:\windows\system32\oleaut32.dll
lsmpoxy	6.1.7600.16385	47.50 KB (48,640 bytes)	7/13/2009 5:17 PM	Microsoft Corporation	c:\windows\system32\lsmpoxy.dll
winlogon	6.1.7600.16447	380.50 KB (389,632 bytes)	3/3/2010 4:45 PM	Microsoft Corporation	c:\windows\system32\winlogon.exe
uxinit	6.1.7600.16385	24.50 KB (25,088 bytes)	7/13/2009 4:54 PM	Microsoft Corporation	c:\windows\system32\uxinit.dll
svchost	6.1.7600.16385	26.50 KB (27,136 bytes)	7/13/2009 4:31 PM	Microsoft Corporation	c:\windows\system32\svchost.exe
umpnpgm	6.1.7600.16385	395.00 KB (404,480 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\umpnpgm.dll
spinif	6.1.7600.16385	103.00 KB (105,472 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\spinif.dll
devrtl	6.1.7600.16385	57.00 KB (58,368 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\devrtl.dll
umpo	6.1.7600.16385	160.00 KB (163,840 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\umpo.dll
setupapi	6.1.7600.16385	1.81 MB (1,899,520 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\setupapi.dll
cfgmgr32	6.1.7600.16385	202.50 KB (207,360 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\cfgmgr32.dll
devobj	6.1.7600.16385	91.00 KB (93,184 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\devobj.dll
rpcss	6.1.7600.16385	497.50 KB (509,440 bytes)	7/13/2009 5:00 PM	Microsoft Corporation	c:\windows\system32\rpcss.dll
wmidcprv	6.1.7600.16385	187.00 KB (191,488 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\wmidcprv.dll
fastprox	6.1.7600.16385	888.00 KB (909,312 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\fastprox.dll

wbemcomn	6.1.7600.16385	517.50 KB (529,920 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbemcomn.dll
ntdsapi	6.1.7600.16385	148.50 KB (152,064 bytes)	7/13/2009 4:54 PM	Microsoft Corporation	c:\windows\system32\ntdsapi.dll
wbemprox	6.1.7600.16385	42.50 KB (43,520 bytes)	7/13/2009 4:46 PM	Microsoft Corporation	c:\windows\system32\wbem\wbemprox.dll
wbemsvc	6.1.7600.16385	63.00 KB (64,512 bytes)	7/13/2009 4:46 PM	Microsoft Corporation	c:\windows\system32\wbem\wbemsvc.dll
wmiutils	6.1.7600.16385	134.00 KB (137,216 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\wmiutils.dll
wintrust	6.1.7600.16493	215.50 KB (220,672 bytes)	10/5/2010 8:37 AM	Microsoft Corporation	c:\windows\system32\wintrust.dll
rpcepmapi	6.1.7600.16385	65.50 KB (67,072 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\rpcepmapi.dll
firewallapi	6.1.7600.16385	730.50 KB (748,032 bytes)	7/13/2009 5:08 PM	Microsoft Corporation	c:\windows\system32\firewallapi.dll
version	6.1.7600.16385	28.50 KB (29,184 bytes)	7/13/2009 4:57 PM	Microsoft Corporation	c:\windows\system32\version.dll
wevtvcs	6.1.7600.16385	1.57 MB (1,646,080 bytes)	7/13/2009 4:49 PM	Microsoft Corporation	c:\windows\system32\wevtvcs.dll
lmhsvc	6.1.7600.16385	23.00 KB (23,552 bytes)	7/13/2009 5:09 PM	Microsoft Corporation	c:\windows\system32\lmhsvc.dll
nrpsrv	6.1.7600.16385	14.50 KB (14,848 bytes)	7/13/2009 5:09 PM	Microsoft Corporation	c:\windows\system32\nrpsrv.dll
dhcpcore	6.1.7600.16385	307.00 KB (314,368 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\dhcpcore.dll
dhcpcore6	6.1.7600.16385	219.00 KB (224,256 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\dhcpcore6.dll
dhcpcsvc6	6.1.7600.16385	53.00 KB (54,272 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\dhcpcsvc6.dll

dhcpcsvc	6.1.7600.16385	85.00 KB (87,040 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\dhcpcsvc.dll
nlaapi	6.1.7600.16385	68.50 KB (70,144 bytes)	7/13/2009 5:09 PM	Microsoft Corporation	c:\windows\system32\nlaapi.dll
rasadhlp	6.1.7600.16385	16.00 KB (16,384 bytes)	7/13/2009 5:10 PM	Microsoft Corporation	c:\windows\system32\rasadhlp.dll
winmr	6.1.7600.16385	28.00 KB (28,672 bytes)	7/13/2009 4:53 PM	Microsoft Corporation	c:\windows\system32\winmr.dll
napinsp	6.1.7600.16385	66.50 KB (68,096 bytes)	7/13/2009 5:10 PM	Microsoft Corporation	c:\windows\system32\napinsp.dll
fwpuclnt	6.1.7600.16385	316.50 KB (324,096 bytes)	7/13/2009 5:09 PM	Microsoft Corporation	c:\windows\system32\fwpuclnt.dll
logonui	6.1.7600.16385	27.00 KB (27,648 bytes)	7/13/2009 4:52 PM	Microsoft Corporation	c:\windows\system32\logonui.exe
authui	6.1.7600.16385	1.84 MB (1,926,144 bytes)	7/13/2009 4:58 PM	Microsoft Corporation	c:\windows\system32\authui.dll
cryptui	6.1.7600.16385	1.02 MB (1,065,984 bytes)	7/13/2009 4:49 PM	Microsoft Corporation	c:\windows\system32\cryptui.dll
comctl32	6.10.7600.16385	1.94 MB (2,030,080 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\winsxs\amd64_microsoft-windows-common-controls_6595b64144ccf1df_6.0.7600.16385_none_fa645303170382f6_comctl32.dll
shlwapi	6.1.7600.16385	439.00 KB (449,536 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\shlwapi.dll
uxtheme	6.1.7600.16385	324.50 KB (332,288 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\uxtheme.dll
gdiplus	6.1.7600.16385	2.06 MB (2,165,248 bytes)	7/13/2009 4:40 PM	Microsoft Corporation	c:\windows\winsxs\amd64_microsoft-windows-gdiplus_6595b64144ccf1df_1.1.7600.16385_none_2b4f45e87195fcc4_gdiplus.dll
dui70	6.1.7600.16385	954.00 KB (976,896 bytes)	7/13/2009 4:41 PM	Microsoft Corporation	c:\windows\system32\dui70.dll

duser 6.1.7600.16385 254.50 KB (260,608 bytes) 7/13/2009 4:39 PM Microsoft Corporation c:\windows\system32\duser.dll

sndvolso 6.1.7600.16385 220.00 KB (225,280 bytes) 7/13/2009 5:19 PM Microsoft Corporation c:\windows\system32\sndvolso.dll

hid 6.1.7600.16385 29.50 KB (30,208 bytes) 7/13/2009 5:06 PM Microsoft Corporation c:\windows\system32\hid.dll

mmdevapi 6.1.7600.16385 277.50 KB (284,160 bytes) 7/13/2009 5:18 PM Microsoft Corporation c:\windows\system32\mmdevapi.dll

propsys 7.0.7600.16385 1.16 MB (1,212,416 bytes) 7/13/2009 4:56 PM Microsoft Corporation c:\windows\system32\propsys.dll

dwmapi 6.1.7600.16385 80.50 KB (82,432 bytes) 7/13/2009 4:37 PM Microsoft Corporation c:\windows\system32\dwmapi.dll

xmlite 1.3.1000.0 195.00 KB (199,680 bytes) 7/13/2009 5:41 PM Microsoft Corporation c:\windows\system32\xmlite.dll

windowscodecs 6.1.7600.16385 1.13 MB (1,189,888 bytes) 7/13/2009 4:42 PM Microsoft Corporation c:\windows\system32\windowscodecs.dll

winbrand 6.1.7600.16385 16.00 KB (16,384 bytes) 7/13/2009 4:30 PM Microsoft Corporation c:\windows\system32\winbrand.dll

vaultcredprovider 6.1.7600.16385 78.50 KB (80,384 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\vaultcredprovider.dll

smartcardcredentialprovider 6.1.7600.16385 185.50 KB (189,952 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\smartcardcredentialprovider.dll

certcredprovider 6.1.7600.16385 126.00 KB (129,024 bytes) 7/13/2009 4:49 PM Microsoft Corporation c:\windows\system32\certcredprovider.dll

raslap 6.1.7600.16385 396.00 KB (405,504 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\raslap.dll

rasapi32 6.1.7600.16385 375.50 KB (384,512 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\rasapi32.dll

rasman 6.1.7600.16385 98.00 KB (100,352 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\rasman.dll

rtutils 6.1.7600.16617 51.00 KB (52,224 bytes) 10/5/2010 8:37 AM Microsoft Corporation c:\windows\system32\rtutils.dll

shacct 6.1.7600.16385 132.00 KB (135,168 bytes) 7/13/2009 4:55 PM Microsoft Corporation c:\windows\system32\shacct.dll

gpsvc 6.1.7600.16385 758.00 KB (776,192 bytes) 7/13/2009 4:54 PM Microsoft Corporation c:\windows\system32\gpsvc.dll

profsvc 6.1.7600.16385 203.50 KB (208,384 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\profsvc.dll

atl 3.5.2284.0 88.50 KB (90,624 bytes) 7/13/2009 5:34 PM Microsoft Corporation c:\windows\system32\atl.dll

dsrole 6.1.7600.16385 32.00 KB (32,768 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\dsrole.dll

sens 6.1.7600.16385 63.00 KB (64,512 bytes) 7/13/2009 4:34 PM Microsoft Corporation c:\windows\system32\sens.dll

schedsvc 6.1.7600.16385 1.05 MB (1,104,384 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\schedsvc.dll

shell32 6.1.7600.16644 13.51 MB (14,162,944 bytes) 10/5/2010 8:37 AM Microsoft Corporation c:\windows\system32\shell32.dll

netapi32 6.1.7600.16385 71.00 KB (72,704 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\netapi32.dll

wkscli 6.1.7600.16385 70.00 KB (71,680 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\wkscli.dll

ktmw32 6.1.7600.16385 22.50 KB (23,040 bytes) 7/13/2009 4:19 PM Microsoft Corporation c:\windows\system32\ktmw32.dll

taskcomp 6.1.7600.16385 462.50 KB (473,600 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\taskcomp.dll

wmisvc 6.1.7600.16385 237.00 KB (242,688 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\wbem\wmisvc.dll

iphlpvc 6.1.7600.16385 552.50 KB (565,760 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\iphlpvc.dll

sqmapi 6.1.7600.16385 229.50 KB (235,008 bytes) 7/13/2009 4:40 PM Microsoft Corporation c:\windows\system32\sqmapi.dll

wdscore 6.1.7600.16385 265.00 KB (271,360 bytes) 7/13/2009 4:28 PM Microsoft Corporation c:\windows\system32\wdscore.dll

srsvc 6.1.7600.16385 230.00 KB (235,520 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\srsvc.dll

browser 6.1.7600.16385 133.00 KB (136,192 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\browser.dll

vssapi 6.1.7600.16385 1.66 MB (1,745,408 bytes) 7/13/2009 4:38 PM Microsoft Corporation c:\windows\system32\vssapi.dll

vsstrace 6.1.7600.16385 75.00 KB (76,800 bytes) 7/13/2009 4:36 PM Microsoft Corporation c:\windows\system32\vsstrace.dll

sscore 6.1.7600.16385 13.00 KB (13,312 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\sscore.dll

clusapi 6.1.7600.16385 307.00 KB (314,368 bytes) 7/13/2009 4:34 PM Microsoft Corporation c:\windows\system32\clusapi.dll

resutils 6.1.7600.16385 84.00 KB (86,016 bytes) 7/13/2009 4:34 PM Microsoft Corporation c:\windows\system32\resutils.dll

netprofm 6.1.7600.16385 449.00 KB (459,776 bytes) 7/13/2009 5:12 PM Microsoft Corporation c:\windows\system32\netprofm.dll

wbemcore 6.1.7600.16385 1.16 MB (1,220,096 bytes) 7/13/2009 4:48 PM Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll

esscli 6.1.7600.16385 430.00 KB (440,320 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\wbem\esscli.dll

nci 6.1.7600.16385 87.50 KB (89,600 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\nci.dll

repdrvfs 6.1.7600.16385 441.00 KB (451,584 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\wbem\repdrvfs.dll

wmiprvsd 6.1.7600.16385 732.50 KB (750,080 bytes) 7/13/2009 4:48 PM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll

ncobjapi 6.1.7600.16385 67.50 KB (69,120 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\ncobjapi.dll

wbemess 6.1.7600.16385 494.00 KB (505,856 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll

npmproxy 6.1.7600.16385 31.00 KB (31,744 bytes) 7/13/2009 5:12 PM Microsoft Corporation c:\windows\system32\npmproxy.dll

certprop 6.1.7600.16385 78.50 KB (80,384 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\certprop.dll

winscard 6.1.7600.16385 212.50 KB (217,600 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\winscard.dll

sessenv 6.1.7600.16385 102.50 KB (104,960 bytes) 7/13/2009 5:17 PM Microsoft Corporation c:\windows\system32\sessenv.dll

ncprov 6.1.7600.16385 76.50 KB (78,336 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll

appmgmts 6.1.7600.16385 189.00 KB (193,536 bytes) 7/13/2009 4:54 PM Microsoft Corporation c:\windows\system32\appmgmts.dll

adslidpc 6.1.7600.16385 231.00 KB (236,544 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\adslidpc.dll

es 2001.12.8530.16385 393.50 KB (402,944 bytes) 7/13/2009 5:00 PM Microsoft Corporation c:\windows\system32\es.dll

nsisvc 6.1.7600.16385 25.00 KB (25,600 bytes) 7/13/2009 4:21 PM Microsoft Corporation c:\windows\system32\nsisvc.dll

w32time 6.1.7600.16385 373.00 KB (381,952 bytes) 7/13/2009 4:49 PM Microsoft Corporation c:\windows\system32\w32time.dll

vmactimeprovider 6.1.7600.16385 50.50 KB (51,712 bytes) 7/13/2009 4:43 PM Microsoft Corporation c:\windows\system32\vmactimeprovider.dll

uxsms 6.1.7600.16385 38.00 KB (38,912 bytes) 7/13/2009 4:37 PM Microsoft Corporation c:\windows\system32\uxsms.dll

trkwks 6.1.7600.16385 117.00 KB (119,808 bytes) 7/13/2009 4:59 PM Microsoft Corporation c:\windows\system32\trkwks.dll

hidserv 6.1.7600.16385 38.00 KB (38,912 bytes) 7/13/2009 5:06 PM Microsoft Corporation c:\windows\system32\hidserv.dll

umrdp 6.1.7600.16385 190.50 KB (195,072 bytes) 7/13/2009 5:18 PM Microsoft Corporation c:\windows\system32\umrdp.dll

winspool 6.1.7600.16385 431.50 KB (441,856 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\winspool.drv

umb 6.1.7600.16385 58.50 KB (59,904 bytes) 7/13/2009 4:35 PM Microsoft Corporation c:\windows\system32\umb.dll

wdi 6.1.7600.16385 88.50 KB (90,624 bytes) 7/13/2009 4:31 PM Microsoft Corporation c:\windows\system32\wdi.dll

netman 6.1.7600.16385 352.00 KB (360,448 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\netman.dll

netshell 6.1.7600.16385 2.53 MB (2,651,136 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\netshell.dll

rasdlg 6.1.7600.16385 840.50 KB (860,672 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\rasdlg.dll

mprapi 6.1.7600.16385 215.50 KB (220,672 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\mprapi.dll

netcfgx 6.1.7600.16385 505.00 KB (517,120 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\netcfgx.dll

hnetcfg 6.1.7600.16385 414.50 KB (424,448 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\hnetcfg.dll

radardt 6.1.7600.16385 95.50 KB (97,792 bytes) 7/13/2009 4:32 PM Microsoft Corporation c:\windows\system32\radardt.dll

dnssrslvr 6.1.7600.16385 178.00 KB (182,272 bytes) 7/13/2009 4:21 PM Microsoft Corporation c:\windows\system32\dnssrslvr.dll

dnsextd 6.1.7600.16385 8.00 KB (8,192 bytes) 7/13/2009 5:12 PM Microsoft Corporation c:\windows\system32\dnsextd.dll

wkssvc 6.1.7600.16385 116.00 KB (118,784 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\wkssvc.dll

cryptsvc 6.1.7600.16385 171.00 KB (175,104 bytes) 7/13/2009 4:49 PM Microsoft Corporation c:\windows\system32\cryptsvc.dll

nlasvc 6.1.7600.16385 295.00 KB (302,080 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\nlasvc.dll

ncsi 6.1.7600.16385 204.50 KB (209,408 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\ncsi.dll

winhttp 6.1.7600.16385 428.50 KB (438,784 bytes) 7/13/2009 5:11 PM Microsoft Corporation c:\windows\system32\winhttp.dll

webio 6.1.7600.16385 385.50 KB (394,752 bytes) 7/13/2009 5:11 PM Microsoft Corporation c:\windows\system32\webio.dll

ssdpapi 6.1.7600.16385 50.00 KB (51,200 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\ssdpapi.dll

esent 6.1.7600.16385 2.45 MB (2,565,120 bytes) 7/13/2009 4:50 PM Microsoft Corporation c:\windows\system32\esent.dll

psapi 6.1.7600.16385 9.00 KB (9,216 bytes) 7/13/2009 4:26 PM Microsoft Corporation c:\windows\system32\psapi.dll

apphostsvc 7.5.7600.16385 64.00 KB (65,536 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetsrv\apphostsvc.dll

iisutil 7.5.7600.16385 274.50 KB (281,088 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetsrv\iisutil.dll

nativerd 7.5.7600.16385 458.50 KB (469,504 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetsrv\nativerd.dll

iisres 7.5.7600.16385 215.00 KB (220,160 bytes) 7/13/2009 5:26 PM Microsoft Corporation c:\windows\system32\inetsrv\iisres.dll

mlang 6.1.7600.16385 221.50 KB (226,816 bytes) 7/13/2009 4:55 PM Microsoft Corporation c:\windows\system32\mlang.dll

termsrv 6.1.7600.16385 690.00
KB (706,560 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\termsrv.dll

icaapi 6.1.7600.16385 22.00 KB
(22,528 bytes) 7/13/2009 5:16 PM
Microsoft Corporation
c:\windows\system32\icaapi.dll

regapi 6.1.7600.16385 92.50 KB
(94,720 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\regapi.dll

tlscsp 6.1.7600.16385 72.00 KB
(73,728 bytes) 7/13/2009 5:16 PM
Microsoft Corporation
c:\windows\system32\tlscsp.dll

rdpcorekmts 6.1.7600.16385 146.00
KB (149,504 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\rdpcorekmts.

dll

rdpwsx 6.1.7600.16385 74.50 KB
(76,288 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\rdpwsx.dll

dps 6.1.7600.16385 159.00
KB (162,816 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\dps.dll

taskschd 6.1.7600.16385 1.11 MB
(1,168,896 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\taskschd.dll

wdiasqmmodule 6.1.7600.16385
35.00 KB (35,840 bytes)
7/13/2009 4:40 PM
Microsoft Corporation
c:\windows\system32\wdiasqmmodule

ule.dll

msdtc 2001.12.8530.16385 138.50
KB (141,824 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtc.exe

msdtctm 2001.12.8530.16385 1.44 MB
(1,509,888 bytes) 7/13/2009 5:00 PM
Microsoft Corporation
c:\windows\system32\msdtctm.dll

msdtcprx 2001.12.8530.16385 728.00
KB (745,472 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtcprx.dll

mtxclu 2001.12.8530.16385 364.00
KB (372,736 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\mtxclu.dll

msdtclog 2001.12.8530.16385 122.00
KB (124,928 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtclog.dll

winmm 6.1.7600.16385 212.50
KB (217,600 bytes) 7/13/2009 5:18 PM
Microsoft Corporation
c:\windows\system32\winmm.dll

xolehlp 2001.12.8530.16385 58.00 KB
(59,392 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\xolehlp.dll

comres 2001.12.8530.16385 1.24 MB
(1,297,408 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\comres.dll

msdtcvsp1res 2001.12.8530.16385
21.00 KB (21,504 bytes)
7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtcvsp1res

.dll

mtxoci 2001.12.8530.16385 153.00
KB (156,672 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\mtxoci.dll

mpr 6.1.7600.16385 79.00 KB
(80,896 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\mpr.dll

drprov 6.1.7600.16385 24.00 KB
(24,576 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\drprov.dll

ntlanman 6.1.7600.16385 126.50
KB (129,536 bytes) 7/13/2009 4:48 PM
Microsoft Corporation
c:\windows\system32\ntlanman.dll

cscapi 6.1.7600.16385 45.00 KB
(46,080 bytes) 7/13/2009 4:24 PM
Microsoft Corporation
c:\windows\system32\cscapi.dll

taskhost 6.1.7600.16385 67.50 KB
(69,120 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\taskhost.exe

msctfmonitor 6.1.7600.16385 27.50 KB
(28,160 bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\msctfmonitor

.dll

msutb 6.1.7600.16385 230.00
KB (235,520 bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\msutb.dll

dimsjob 6.1.7600.16385 39.50 KB
(40,448 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\dimsjob.dll

rdpclip 6.1.7600.16385 204.50
KB (209,408 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\rdpclip.exe

dwm 6.1.7600.16385 117.50
KB (120,320 bytes) 7/13/2009 4:37 PM
Microsoft Corporation
c:\windows\system32\dwm.exe

dwmredir 6.1.7600.16385 125.50
KB (128,512 bytes) 7/13/2009 4:37 PM
Microsoft Corporation
c:\windows\system32\dwmredir.dll

dwmcore 6.1.7600.16385 1.56 MB
(1,634,304 bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\dwmcore.dll

d3d10_1 6.1.7600.16385 192.50
KB (197,120 bytes) 7/13/2009 4:41 PM
Microsoft Corporation
c:\windows\system32\d3d10_1.dll

d3d10_1core 6.1.7600.16385 311.50
KB (318,976 bytes) 7/13/2009 4:41 PM
Microsoft Corporation
c:\windows\system32\d3d10_1core.

dll

dxgi 6.1.7600.16385 643.00
KB (658,432 bytes) 7/13/2009 4:41 PM
Microsoft Corporation
c:\windows\system32\dxgi.dll

explorer 6.1.7600.16450 2.74 MB
(2,870,272 bytes) 3/3/2010 4:45 PM
Microsoft Corporation
c:\windows\explorer.exe

explorerframe 6.1.7600.16385
1.78 MB (1,863,680 bytes)
7/13/2009 4:57 PM
Microsoft Corporation
c:\windows\system32\explorerfram

e.dll

powrprof 6.1.7600.16385 163.50
KB (167,424 bytes) 7/13/2009 4:27 PM
Microsoft Corporation
c:\windows\system32\powrprof.dll

ehstorshell 6.1.7600.16385 198.50
KB (203,264 bytes) 7/13/2009 5:00 PM
Microsoft Corporation
c:\windows\system32\ehstorshell.dll

l

ntshrui 6.1.7600.16385 498.00
KB (509,952 bytes) 7/13/2009 4:57 PM
Microsoft Corporation
c:\windows\system32\ntshrui.dll

iconcodecservice 6.1.7600.16385
14.00 KB (14,336 bytes)
7/13/2009 4:37 PM
Microsoft Corporation
c:\windows\system32\iconcodecserv

vice.dll

timedate 6.1.7600.16385 503.00
KB (515,072 bytes) 7/13/2009 4:56 PM
Microsoft Corporation
c:\windows\system32\timedate.cpl

shdocvw 6.1.7600.16385 191.50
KB (196,096 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\system32\shdocvw.dll

linkinfo 6.1.7600.16385 29.00 KB
(29,696 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\system32\linkinfo.dll

msftedit 5.41.21.2509 781.00 KB (799,744
bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\msftedit.dll

mssl31	3.10.349.0	217.00 KB (222,208 bytes)	7/13/2009 4:39 PM	Microsoft Corporation	c:\windows\system32\mssl31.dll
urlmon	8.0.7600.16625	1.43 MB (1,494,528 bytes)	10/5/2010 8:37 AM	Microsoft Corporation	c:\windows\system32\urlmon.dll
iertutil	8.0.7600.16385	2.33 MB (2,440,704 bytes)	7/13/2009 4:59 PM	Microsoft Corporation	c:\windows\system32\iertutil.dll
ieframe	8.0.7600.16625	11.79 MB (12,364,800 bytes)	10/5/2010 8:37 AM	Microsoft Corporation	c:\windows\system32\ieframe.dll
oleacc	7.0.0.0	324.00 KB (331,776 bytes)	7/13/2009 4:39 PM	Microsoft Corporation	c:\windows\system32\oleacc.dll
stobject	6.1.7600.16385	250.00 KB (256,000 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\stobject.dll
batmeter	6.1.7600.16385	730.50 KB (748,032 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\batmeter.dll
prnfdrr	6.1.7600.16385	407.00 KB (416,768 bytes)	7/13/2009 5:40 PM	Microsoft Corporation	c:\windows\system32\prnfdrr.dll
dxdp	6.1.7600.16385	449.00 KB (459,776 bytes)	7/13/2009 5:21 PM	Microsoft Corporation	c:\windows\system32\dxdp.dll
syncreg	2007.94.7600.16385	72.00 KB (73,728 bytes)	7/13/2009 5:22 PM	Microsoft Corporation	c:\windows\system32\syncreg.dll
pnidui	6.1.7600.16385	1.72 MB (1,807,872 bytes)	7/13/2009 5:08 PM	Microsoft Corporation	c:\windows\system32\pnidui.dll
qutil	6.1.7600.16385	105.00 KB (107,520 bytes)	7/13/2009 5:07 PM	Microsoft Corporation	c:\windows\system32\qutil.dll
actioncenter	6.1.7600.16385	762.50 KB (780,800 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\actioncenter.dll
imapi2	6.1.7600.16385	493.50 KB (505,344 bytes)	7/13/2009 5:01 PM	Microsoft Corporation	c:\windows\system32\imapi2.dll
qagent	6.1.7600.16385	259.00 KB (265,216 bytes)	7/13/2009 5:07 PM	Microsoft Corporation	c:\windows\system32\qagent.dll

hgcppl	6.1.7600.16385	324.50 KB (332,288 bytes)	7/13/2009 4:57 PM	Microsoft Corporation	c:\windows\system32\hgcppl.dll
actxprxy	6.1.7600.16385	936.50 KB (958,976 bytes)	7/13/2009 5:41 PM	Microsoft Corporation	c:\windows\system32\actxprxy.dll
ieproxy	8.0.7600.16625	440.50 KB (451,072 bytes)	10/5/2010 8:37 AM	Microsoft Corporation	c:\program files\internet explorer\ieproxy.dll
thumbcache	6.1.7600.16385	110.50 KB (113,152 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\thumbcache.dll
networkexplorer	6.1.7600.16385	1.60 MB (1,672,704 bytes)	7/13/2009 5:08 PM	Microsoft Corporation	c:\windows\system32\networkexplorer.dll
structuredquery	7.0.7600.16385	472.50 KB (483,840 bytes)	7/13/2009 5:29 PM	Microsoft Corporation	c:\windows\system32\structuredquery.dll
wercnpl	6.1.7600.16385	1.22 MB (1,280,512 bytes)	7/13/2009 4:41 PM	Microsoft Corporation	c:\windows\system32\wercnpl.dll
wer	6.1.7600.16385	473.00 KB (484,352 bytes)	7/13/2009 4:41 PM	Microsoft Corporation	c:\windows\system32\wer.dll
framedynos	6.1.7600.16385	288.50 KB (295,424 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\framedynos.dll
wercplsupport	6.1.7600.16385	82.50 KB (84,480 bytes)	7/13/2009 4:40 PM	Microsoft Corporation	c:\windows\system32\wercplsupport.dll
msxml6	6.30.7600.16385	1.91 MB (1,999,360 bytes)	7/13/2009 5:43 PM	Microsoft Corporation	c:\windows\system32\msxml6.dll
hcproviders	6.1.7600.16385	30.50 KB (31,232 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\hcproviders.dll
searchfolder	6.1.7600.16385	845.00 KB (865,280 bytes)	7/13/2009 4:59 PM	Microsoft Corporation	c:\windows\system32\searchfolder.dll
aclui	6.1.7600.16385	150.50 KB (154,112 bytes)	7/13/2009 4:57 PM	Microsoft Corporation	c:\windows\system32\aclui.dll

rarext64	Not Available	42.00 KB (43,008 bytes)	11/2/2010 7:34 AM	Not Available	c:\program files\winrar\rarext64.dll
wininet	8.0.7600.16625	1.14 MB (1,192,960 bytes)	10/5/2010 8:37 AM	Microsoft Corporation	c:\windows\system32\wininet.dll
normaliz	6.1.7600.16385	2.50 MB (2,560 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\normaliz.dll
zipfldr	6.1.7600.16385	357.50 KB (366,080 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\zipfldr.dll
rstrtmgr	6.1.7600.16385	184.00 KB (188,416 bytes)	7/13/2009 4:35 PM	Microsoft Corporation	c:\windows\system32\rstrtmgr.dll
twext	6.1.7600.16385	168.50 KB (172,544 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\twext.dll
acppage	6.1.7600.16385	52.00 KB (53,248 bytes)	7/13/2009 4:40 PM	Microsoft Corporation	c:\windows\system32\acppage.dll
sfc	6.1.7600.16385	3.00 KB (3,072 bytes)	7/13/2009 4:25 PM	Microsoft Corporation	c:\windows\system32\sfc.dll
sfc_os	6.1.7600.16385	44.00 KB (45,056 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\sfc_os.dll
msi	5.0.7600.16385	3.06 MB (3,211,776 bytes)	7/13/2009 4:51 PM	Microsoft Corporation	c:\windows\system32\msi.dll
taskmgr	6.1.7600.16385	251.00 KB (257,024 bytes)	7/13/2009 4:31 PM	Microsoft Corporation	c:\windows\system32\taskmgr.exe
credui	6.1.7600.16385	213.00 KB (218,112 bytes)	7/13/2009 4:53 PM	Microsoft Corporation	c:\windows\system32\credui.dll
utildll	6.1.7600.16385	34.00 KB (34,816 bytes)	7/13/2009 5:17 PM	Microsoft Corporation	c:\windows\system32\utildll.dll
browcli	6.1.7600.16385	57.00 KB (58,368 bytes)	7/13/2009 4:53 PM	Microsoft Corporation	c:\windows\system32\browcli.dll
cmd	6.1.7600.16385	336.50 KB (344,576 bytes)	7/13/2009 4:34 PM	Microsoft Corporation	c:\windows\system32\cmd.exe

conhost 6.1.7600.16385 330.50
KB (338,432 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\conhost.exe

sqlservr 2005.90.4267.0 36.94
MB (38,734,168 bytes) 10/16/2009 11:23 PM
Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe

msvc80 8.0.50727.4927 783.81
KB (802,624 bytes) 7/13/2009 1:37 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft
t.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4927_n
one_88dce9872fb18caf\msvc80.dll

msvc80 8.0.50727.4927 1.02 MB
(1,068,352 bytes) 7/13/2009 1:37 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft
t.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4927_n
one_88dce9872fb18caf\msvc80.dll

sqlos 2005.90.4035.0 15.84 KB
(16,224 bytes) 11/25/2008 11:45 AM
Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlos.dll

opends60 2005.90.1399.0 22.21 KB
(22,744 bytes) 10/14/2005 3:31 PM
Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\opends60.dll

instapi 2005.90.1399.0 40.71 KB
(41,688 bytes) 10/14/2005 3:23 PM
Microsoft Corporation
c:\program files\microsoft sql
server\90\shared\instapi.dll

sqllevn70 2005.90.4035.0 1.66 MB
(1,738,080 bytes) 11/25/2008 11:40 AM
Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\resources\1033\sqllev
n70.rll

mscoree 4.0.31106.0 434.33 KB (444,752
bytes) 10/5/2010 8:40 AM
Microsoft Corporation
c:\windows\system32\mscoree.dll

security 6.1.7600.16385 5.00 KB
(5,120 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\security.dll

msfte 12.0.9732.0 3.64 MB (3,813,744
bytes) 6/17/2009 6:00 AM
Microsoft Corporation
c:\program files\microsoft sql
server\mssql.1\mssql\bin\msfte.dll

dbghelp 6.6.7.5 1.27 MB (1,326,944
bytes) 11/25/2008 11:38 AM
Microsoft Corporation
c:\program files\microsoft sql
server\90\shared\dbghelp.dll

sqlncli 2005.90.4035.0 2.75 MB
(2,883,424 bytes) 11/25/2008 11:45 AM
Microsoft Corporation
c:\windows\system32\sqlncli.dll

comctl32 5.82.7600.16385 619.00
KB (633,856 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft
t.windows.common-
controls_6595b64144ccf1df_5.82.7600.16385_no
ne_a44af8ec57f961cf\comctl32.dll

comdlg32 6.1.7600.16385 581.50
KB (595,456 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\system32\comdlg32.dll

sqlnclir 2005.90.1399.0 201.21
KB (206,040 bytes) 10/14/2005 3:31 PM
Microsoft Corporation
c:\windows\system32\sqlnclir.rll

netbios 6.1.7600.16385 18.50 KB
(18,944 bytes) 7/13/2009 5:09 PM
Microsoft Corporation
c:\windows\system32\netbios.dll

msinfo32 6.1.7600.16385 370.00
KB (378,880 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\msinfo32.exe

mfc42u 6.6.8063.0 1.29 MB (1,357,312
bytes) 7/13/2009 5:35 PM
Microsoft Corporation
c:\windows\system32\mfc42u.dll

odbc32 6.1.7600.16385 696.00
KB (712,704 bytes) 7/13/2009 5:29 PM
Microsoft Corporation
c:\windows\system32\odbc32.dll

odbcint 6.1.7600.16385 224.00
KB (229,376 bytes) 7/13/2009 5:28 PM
Microsoft Corporation
c:\windows\system32\odbcint.dll

ehstorapi 6.1.7600.16385 141.50
KB (144,896 bytes) 7/13/2009 5:00 PM
Microsoft Corporation
c:\windows\system32\ehstorapi.dll

wmiprvse 6.1.7600.16385 360.00
KB (368,640 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\wmipr
vse.exe

wmiperfclass 6.1.7600.16385 133.00
KB (136,192 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\wbem\wmipe

rfclass.dll

pdh 6.1.7600.16385 293.00
KB (300,032 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\pdh.dll

cimwin32 6.1.7600.16385 1.96 MB
(2,055,168 bytes) 7/13/2009 4:48 PM
Microsoft Corporation
c:\windows\system32\wbem\cimwi

n32.dll

schedcli 6.1.7600.16385 23.50 KB
(24,064 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\schedcli.dll

wmi 6.1.7600.16385 5.00 KB
(5,120 bytes) 7/13/2009 5:41 PM
Microsoft Corporation
c:\windows\system32\wmi.dll

ntevt 6.1.7600.16385 260.00
KB (266,240 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\ntevt.
dll

provthrd 6.1.7600.16385 300.00
KB (307,200 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\provthrd.dll

msvcirt 7.0.7600.16385 76.50 KB
(78,336 bytes) 7/13/2009 4:18 PM
Microsoft Corporation
c:\windows\system32\msvcirt.dll

wsock32 6.1.7600.16385 18.00 KB
(18,432 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\wsock32.dll

tapi32 6.1.7600.16385 243.00
KB (248,832 bytes) 7/13/2009 5:41 PM
Microsoft Corporation
c:\windows\system32\tapi32.dll

[Services]

Display Name	Name	State
Start Mode	Service Type	Path
Error Control	Start Name	Tag ID
Application Experience	AeLookupSvc	Stopped
Disabled	Share Process	
	c:\windows\system32\svchost.exe -	
k netsvcs	Normal	localSystem 0
Application Layer Gateway Service	ALG	Stopped
Manual	Share Process	Own
Process	c:\windows\system32\alg.exe	
Normal	NT	
AUTHORITY\LocalService	0	
Application Host Helper Service	AppHostSvc	Running
AppHostSvc	Running	Auto
Share Process		
	c:\windows\system32\svchost.exe -	
k apphost	Normal	LocalSystem 0
Application Identity	AppIDSvc	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k localserviceandnoimpersonation	Normal	NT Authority\LocalService 0
Application Information	Appinfo	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k netsvcs	Normal	LocalSystem 0
Application Management	AppMgmt	Running
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k netsvcs	Normal	LocalSystem 0
Windows Audio Endpoint Builder	AudioEndpointBuilder	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k localsystemnetworkrestricted	Normal	LocalSystem 0
Windows Audio	AudioSrv	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k localservicenetworkrestricted	Normal	NT AUTHORITY\LocalService 0
Base Filtering Engine	BFE	Stopped
Disabled	Share Process	
	c:\windows\system32\svchost.exe -	
k localservicenetwork	Normal	NT AUTHORITY\LocalService 0

Background Intelligent Transfer Service BITS
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Computer Browser Browser Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Certificate Propagation CertPropSvc Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

HP Smart Array SAS/SATA Event Notification
 Service Cissesrv Stopped Disabled
 Own Process "c:\program
 files\hp\cissesrv\cissesrv.exe" Normal
 LocalSystem 0

Microsoft .NET Framework NGEN v2.0.50727_X86
 clr_optimization_v2.0.50727_32
 Stopped Manual Own
 Process c:\windows\microsoft.net\framework
 k\v2.0.50727\mscorsvw.exe Ignore
 LocalSystem 0

Microsoft .NET Framework NGEN v2.0.50727_X64
 clr_optimization_v2.0.50727_64
 Stopped Manual Own
 Process c:\windows\microsoft.net\framework
 k64\v2.0.50727\mscorsvw.exe Ignore
 LocalSystem 0

COM+ System Application COMSysApp Stopped
 Manual Own Process
 c:\windows\system32\dlhhost.exe
 /processid:{02d4b3f1-fd88-11d1-960d-
 00805fc79235} Normal
 LocalSystem 0

HP ProLiant iLO 3 Remote JML Service
 CpqRcmc3 Stopped Disabled
 Share Process
 "c:\program files\hewlett-
 packard\ilo 3\service\proliantmonitor.exe"
 Normal LocalSystem 0

Cryptographic Services CryptSvc Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 Authority\NetworkService 0

DCOM Server Process Launcher
 DcomLaunch Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k dcomlaunch Normal
 LocalSystem 0

Disk Defragmenter defragsvc Stopped
 Manual Own Process
 c:\windows\system32\svchost.exe -
 k defragsvc Normal LocalSystem 0

DHCP Client Dhcp Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 NT Authority\LocalService 0

DNS Client Dnscache Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 Authority\NetworkService 0

Wired AutoConfig dot3svc Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 Diagnostic Policy System DPS Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetwork Normal
 NT
 Authority\LocalService 0

Extensible Authentication Protocol EapHost
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Encrypting File System (EFS) EFS
 Running Auto Share
 Process c:\windows\system32\lsass.exe
 Normal LocalSystem 0

Emulex HBA Discovery Emulex HBA Discovery
 Stopped Manual Own
 Process c:\program files
 (x86)\emulex\util\common\hbadiscsrvr.exe
 Normal LocalSystem 0

Emulex HBA Management Emulex HBA
 Management Stopped Disabled Own
 Process c:\program files
 (x86)\emulex\util\common\rmserver.exe Normal
 LocalSystem 0

Emulex MILI Management Emulex Management
 Interface Library v2 Stopped Manual
 Own Process c:\program files
 (x86)\emulex\util\common\mili2service.exe
 Normal LocalSystem 0

Emulex SvcMgr Emulex SvcMgr
 Stopped Disabled Own
 Process c:\program files
 (x86)\emulex\util\common\hbahsmgr.exe
 Normal LocalSystem 0

Windows Event Log eventlog Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 NT Authority\LocalService
 0

COM+ Event System EventSystem Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservice Normal NT
 Authority\LocalService 0

Microsoft Fibre Channel Platform Registration
 Service FCRegSvc Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 NT Authority\LocalService
 0

Function Discovery Provider Host fdPHost
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k localservice Normal NT
 Authority\LocalService 0

Function Discovery Resource Publication
 FDResPub Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k localserviceandnoimpersonation Normal
 NT Authority\LocalService
 0

Windows Font Cache Service
 FontCache Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k localserviceandnoimpersonation Normal
 NT Authority\LocalService
 0

Windows Presentation Foundation Font Cache
 3.0.0.0 FontCache3.0.0.0 Stopped
 Manual Own Process
 c:\windows\microsoft.net\framework
 k64\v3.0\wpf\presentationfontcache.exe Normal
 NT Authority\LocalService 0

Group Policy Client gpsvc Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Brocade HCM Agent Service
 hcmagent Stopped Manual
 Own Process "c:\program files
 (x86)\brocade\driverpackage\bfa\util\hbaagent\bi
 n\hcmagent.exe" -d -c "c:\program files
 (x86)\brocade\driverpackage\bfa\util\hbaagent\c
 onf\abyss.conf" Normal
 LocalSystem 0

Human Interface Device Access hidserv
 Running Manual Share
 Process c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 LocalSystem 0

Health Key and Certificate Management hkmsvc
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Windows CardSpace idsvc Stopped
 Manual Share Process
 "c:\windows\microsoft.net\framework
 rk64\v3.0\windows communication
 foundation\infocard.exe" Normal
 LocalSystem 0

IKE and AuthIP IPsec Keying Modules IKEEXT
 Stopped Auto Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

PnP-X IP Bus Enumerator IPBusEnum Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 LocalSystem 0

IP Helper iphlpsvc Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

CNG Key Isolation KeyIso Stopped
 Manual Share Process
 c:\windows\system32\lsass.exe
 Normal LocalSystem 0

KtmRm for Distributed Transaction Coordinator
 KtmRm Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k networkserviceandnoimpersonation Normal
 NT Authority\NetworkService
 0

Server LanmanServer Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Workstation LanmanWorkstation Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 AUTHORITY\NetworkService 0

Link-Layer Topology Discovery Mapper lltdsvc
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k localservice Normal NT
 AUTHORITY\LocalService 0

TCP/IP NetBIOS Helper lmhosts Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 NT AUTHORITY\LocalService
 0

Multimedia Class SchedulerMMCSS Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Windows Firewall MpsSvc Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetwork Normal NT
 Authority\LocalService 0

Distributed Transaction Coordinator MSDTC
 Running Auto Own
 Process c:\windows\system32\msdtc.exe
 Normal NT
 AUTHORITY\NetworkService 0

SQL Server FullText Search (MSSQLSERVER)
 msftesql Stopped Disabled
 Own Process "c:\program
 files\microsoft sql
 server\mssql.1\mssql\binn\msftesql.exe" -
 s:mssql.1 -f:mssqlserver Normal
 LocalSystem 0

Microsoft iSCSI Initiator Service MSISCSI
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Windows Installer msiserver Stopped
 Manual Own Process
 c:\windows\system32\msiexec.exe
 /v Normal LocalSystem 0

SQL Server (MSSQLSERVER)
 MSSQLSERVER Stopped
 Manual Own Process
 "c:\program files\microsoft sql
 server\mssql.1\mssql\binn\sqlservr.exe" -
 smssqlserver Normal LocalSystem 0

SQL Server Active Directory Helper
 MSSQLServerADHelper Stopped
 Disabled Own Process
 "c:\program files\microsoft sql
 server\90\shared\sqladhelp90.exe" Normal
 NT AUTHORITY\NetworkService
 0

Network Access Protection Agent
 napagent Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 AUTHORITY\NetworkService 0

Netlogon Netlogon Stopped Manual
 Share Process
 c:\windows\system32\sass.exe
 Normal LocalSystem 0

Network Connections Netman Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localsystemnetworkrestricted Normal
 LocalSystem 0

Network List Service netprofm Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localservice Normal NT
 AUTHORITY\LocalService 0

Net.Tcp Port Sharing Service
 NetTcpPortSharing Stopped
 Disabled Share Process
 "c:\windows\microsoft.net\frameworko
 rk64\v3.0\windows communication
 foundation\smsvchost.exe" Normal NT
 AUTHORITY\LocalService 0

Network Location Awareness NlaSvc
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 AUTHORITY\NetworkService 0

Network Store Interface Service nsi
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k localservice Normal NT
 Authority\LocalService 0

ONC/RPC Portmapper oncpportmap Stopped
 Manual Own Process
 c:\program files (x86)\qlogic
 corporation\sansurfer\portmap.exe Normal
 LocalSystem 0

Office Source Engine ose Stopped
 Manual Own Process
 "c:\program files (x86)\common
 files\microsoft shared\source engine\ose.exe"
 Normal LocalSystem 0

Performance Counter DLL Host PerfHost
 Stopped Manual Own
 Process c:\windows\syswow64\perfhst.exe
 Normal NT
 AUTHORITY\LocalService 0

Performance Logs & Alerts pla Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetwork Normal NT
 AUTHORITY\LocalService 0

Plug and PlayPlugPlay Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k dcomlaunch Normal
 LocalSystem 0

IPsec Policy Agent PolicyAgent Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k networkservicenetworkrestricted Normal
 NT Authority\NetworkService
 0

Power Power Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k dcomlaunch Normal
 LocalSystem 0

User Profile Service ProfSvc Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

HP ProLiant Health Monitor Service
 ProLiantMonitor Stopped
 Disabled Share Process
 "c:\program files\hewlett-
 packard\ilo 3\service\proliantmonitor.exe"
 Normal LocalSystem 0

Protected Storage ProtectedStorage
 Stopped Manual Share
 Process c:\windows\system32\sass.exe
 Normal LocalSystem 0

QLogic Management Suite Java Agent
 QLManagementAgentJava Stopped
 Auto Own Process
 c:\progra~2\qlogic~1\sansur~1\qlr
 emote.exe Normal LocalSystem 0

Remote Access Auto Connection Manager
 RasAuto Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Remote Access Connection Manager RasMan
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Routing and Remote Access
 RemoteAccess Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Remote Registry RemoteRegistry
 Stopped Disabled Share
 Process c:\windows\system32\svchost.exe -
 k regsvc Normal NT
 AUTHORITY\LocalService 0

RPC Endpoint Mapper RpcEptMapper
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k rpcss Normal NT
 AUTHORITY\NetworkService 0

Remote Procedure Call (RPC) Locator
 RpcLocator Stopped Manual
 Own Process
 c:\windows\system32\locator.exe
 Normal NT
 AUTHORITY\NetworkService 0

Remote Procedure Call (RPC) RpcSs
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k rpcss Normal NT
 AUTHORITY\NetworkService 0

Resultant Set of Policy Provider
 RSoPProv Stopped Manual
 Share Process
 c:\windows\system32\rsopprov.exe
 Normal LocalSystem 0

Special Administration Console Helper sacsvr
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Security Accounts ManagerSamSs Running
 Auto Share Process
 c:\windows\system32\sass.exe
 Normal LocalSystem 0

Smart Card	SCardSvr	Stopped	Manual
	Share Process		
	c:\windows\system32\svchost.exe -		
k localserviceandnoimpersonation	Normal		
	NT AUTHORITY\LocalService		
	0		
Task Scheduler	Schedule	Running	
	Auto	Share Process	
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
Smart Card Removal Policy	SCPolicySvc	Stopped	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
Secondary Logon	seclogon	Stopped	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
System Event Notification Service	SENS	Running	Auto
	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
Remote Desktop Configuration	SessionEnv	Running	Manual
	Share Process		
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
Internet Connection Sharing (ICS)	SharedAccess	Stopped	
	Disabled	Share Process	
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0
Shell Hardware Detection	ShellHWDetection	Stopped	Disabled
	Share		
Process	c:\windows\system32\svchost.exe -		
k netsvcs	Ignore	LocalSystem	0
SNMP Trap	SNMPTRAP	Stopped	Manual
	Own Process		
	c:\windows\system32\snmptrap.exe		
	Normal	NT	
	AUTHORITY\LocalService	0	
Print Spooler	Spooler	Stopped	Disabled
	Own Process		
	c:\windows\system32\spoolsv.exe		
	Normal	LocalSystem	0
Software Protection	sppsvc	Stopped	
	Auto	Own Process	
	c:\windows\system32\sppsvc.exe		
	Normal	NT	
	AUTHORITY\NetworkService	0	
SPP Notification Service	sppuinotify	Stopped	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k localservice	Normal	NT	
	AUTHORITY\LocalService	0	
SQL Server Browser	SQLBrowser	Stopped	
	Manual	Own Process	
	"c:\program files (x86)\microsoft		
sql server\90\shared\sqlbrowser.exe"	Normal	LocalSystem	0

SQL Server Agent (MSSQLSERVER)	SQLSERVERAGENT	Stopped	
	Manual	Own Process	
	"c:\program files\microsoft sql		
server\mssql.1\mssql\bin\sqlagent90.exe" -i	Normal	LocalSystem	0
mssqlserver	Normal	LocalSystem	0
SQL Server VSS Writer	SQLWriter	Stopped	
	Disabled	Own Process	
	"c:\program files\microsoft sql		
server\90\shared\sqlwriter.exe"	Normal	LocalSystem	0
SSDP Discovery	SSDPSRV	Stopped	
	Disabled	Share Process	
	c:\windows\system32\svchost.exe -		
k localserviceandnoimpersonation	Normal	NT AUTHORITY\LocalService	0
Secure Socket Tunneling Protocol Service	SstpSvc	Stopped	Manual
	Share Process		
	c:\windows\system32\svchost.exe -		
k localservice	Normal	NT	
	Authority\LocalService	0	
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual
	Own Process		
	c:\windows\system32\svchost.exe -		
k swprv	Normal	LocalSystem	0
HP ProLiant System Shutdown Service	sysdown	Stopped	Auto
	Share		
Process	"c:\program files\hewlett-		
packard\ilo 3\service\proliantmonitor.exe"	Normal	LocalSystem	0
Telephony	TapiSrv	Stopped	Manual
	Own Process		
	c:\windows\system32\svchost.exe -		
k tapisrv	Normal	NT	
	AUTHORITY\NetworkService	0	
TPM Base Services	TBS	Stopped	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k localserviceandnoimpersonation	Normal	NT AUTHORITY\LocalService	0
Remote Desktop Services	TermService	Running	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k termsvcs	Normal	NT	
	Authority\NetworkService	0	
Thread Ordering Server	THREADORDER	Stopped	Manual
	Share		
Process	c:\windows\system32\svchost.exe -		
k localservice	Normal	NT	
	AUTHORITY\LocalService	0	
Distributed Link Tracking Client	TrkWks	Running	Auto
	Share		
Process	c:\windows\system32\svchost.exe -		
k localsystemnetworkrestricted	Normal	LocalSystem	0
Windows Modules Installer	TrustedInstaller	Stopped	Manual
	Own		
Process	c:\windows\servicing\trustedinstalle		
r.exe	Normal	LocalSystem	0
Interactive Services Detection	UI0Detect	Stopped	Manual
	Own Process		
	c:\windows\system32\ui0detect.exe		
	Normal	LocalSystem	0

Remote Desktop Services UserMode Port	UmRdpService	Running	
Redirector	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k localsystemnetworkrestricted	Normal	NT AUTHORITY\LocalService	0
UPnP Device Host	ppnpshost	Stopped	
	Disabled	Share Process	
	c:\windows\system32\svchost.exe -		
k localserviceandnoimpersonation	Normal	NT AUTHORITY\LocalService	0
Desktop Window Manager Session Manager	UxSms	Running	Auto
	Share Process		
	c:\windows\system32\svchost.exe -		
k localsystemnetworkrestricted	Normal	LocalSystem	0
Credential Manager	VaultSvc	Stopped	
	Manual	Share Process	
	c:\windows\system32\lsass.exe		
	Normal	LocalSystem	0
Virtual Disk	vds	Stopped	Manual
	Own Process		
	c:\windows\system32\vds.exe		
	Normal	LocalSystem	0
Volume Shadow Copy	VSS	Stopped	
	Manual	Own Process	
	c:\windows\system32\vssvc.exe		
	Normal	LocalSystem	0
Windows Time	W32Time	Running	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k localservice	Normal	NT	
	AUTHORITY\LocalService	0	
Windows Process Activation Service	WAS	Stopped	Manual
	Share		
Process	c:\windows\system32\svchost.exe -		
k iissvcs	Normal	LocalSystem	0
Block Level Backup Engine Service	wbengine	Stopped	Manual
	Own Process		
	"c:\windows\system32\wbengine.ex		
e"	Normal	LocalSystem	0
Windows Color System	WcsPlugInService	Stopped	Manual
	Share		
Process	c:\windows\system32\svchost.exe -		
k wccsvc	Normal	NT	
	AUTHORITY\LocalService	0	
Diagnostic Service Host	WdiServiceHost	Stopped	Manual
	Share		
Process	c:\windows\system32\svchost.exe -		
k localservice	Normal	NT	
	AUTHORITY\LocalService	0	
Diagnostic System Host	WdiSystemHost	Running	Manual
	Share		
Process	c:\windows\system32\svchost.exe -		
k localsystemnetworkrestricted	Normal	LocalSystem	0
Windows Event Collector	Weccsvc	Stopped	
	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k networkservice	Normal	NT	
	AUTHORITY\NetworkService	0	
Problem Reports and Solutions Control Panel	werclpsupport	Stopped	
Support	Manual	Share Process	
	c:\windows\system32\svchost.exe -		
k netsvcs	Normal	LocalSystem	0

```

Windows Error Reporting Service WerSvc
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k wersvcgroup Ignore localSystem 0
WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -
k localservice Normal NT
AUTHORITY\LocalService 0
Windows Management Instrumentation
Winmgmt Running Auto
Share Process
c:\windows\system32\svchost.exe -
k netsvcs Ignore localSystem 0

Windows Remote Management (WS-
Management) WinRM Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -
k networkservice Normal NT
AUTHORITY\NetworkService 0

WMI Performance Adapter wmiApSrv Stopped
Manual Own Process
c:\windows\system32\wbem\wmiap
srv.exe Normal localSystem 0

Portable Device Enumerator Service
WPDBusEnum Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
LocalSystem 0
Windows Update wuauserv Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Windows Driver Foundation - User-mode Driver
Framework wudfsvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
LocalSystem 0

[Program Groups]

Group Name Name User Name Default
Start Menu Default:Start Menu Default

Start Menu\Programs Default:Start
Menu\Programs Default
Start Menu\Programs\Accessories
Default:Start
Menu\Programs\Accessories Default

Start Menu\Programs\Accessories\Accessibility
Default:Start
Menu\Programs\Accessories\Accessibility
Default
Start Menu\Programs\Accessories\System Tools
Default:Start
Menu\Programs\Accessories\System Tools
Default
Start Menu\Programs\Maintenance
Default:Start
Menu\Programs\Maintenance Default

Start Menu Public:Start Menu Public

Start Menu\Programs Public:Start
Menu\Programs Public

```

```

Start Menu\Programs\Accessories
Public:Start
Menu\Programs\Accessories Public
Start Menu\Programs\Accessories\Accessibility
Public:Start
Menu\Programs\Accessories\Accessibility
Public
Start Menu\Programs\Accessories\System Tools
Public:Start
Menu\Programs\Accessories\System Tools
Public
Start Menu\Programs\Accessories\Windows
PowerShell Public:Start
Menu\Programs\Accessories\Windows PowerShell
Public
Start Menu\Programs\Administrative Tools
Public:Start
Menu\Programs\Administrative Tools Public

Start Menu\Programs\Administrative
Tools\Terminal Services Public:Start
Menu\Programs\Administrative Tools\Terminal
Services Public
Start Menu\Programs\Brocade Adapter Software
Public:Start
Menu\Programs\Brocade Adapter Software
Public
Start Menu\Programs\Debugging Tools for
Windows (x64) Public:Start
Menu\Programs\Debugging Tools for Windows
(x64) Public
Start Menu\Programs\Emulex
Public:Start Menu\Programs\Emulex
Public
Start Menu\Programs\HP System Tools
Public:Start Menu\Programs\HP
System Tools Public
Start Menu\Programs\HP System Tools\HP Array
Configuration Utility Public:Start
Menu\Programs\HP System Tools\HP Array
Configuration Utility Public
Start Menu\Programs\Maintenance
Public:Start
Menu\Programs\Maintenance Public

Start Menu\Programs\Microsoft SQL Server 2005
Public:Start
Menu\Programs\Microsoft SQL Server 2005
Public
Start Menu\Programs\Microsoft SQL Server
2005\Configuration Tools Public:Start
Menu\Programs\Microsoft SQL Server
2005\Configuration Tools Public
Start Menu\Programs\NeoSmart Technologies
Public:Start
Menu\Programs\NeoSmart Technologies Public

Start Menu\Programs\NeoSmart
Technologies\EasyBCD Public:Start
Menu\Programs\NeoSmart
Technologies\EasyBCD Public
Start Menu\Programs\Startup
Public:Start Menu\Programs\Startup
Public
Start Menu\Programs\WinRAR
Public:Start
Menu\Programs\WinRAR Public
Start Menu SQLWESTMERE\guz:Start Menu
SQLWESTMERE\guz
Start Menu\Programs
SQLWESTMERE\guz:Start
Menu\Programs SQLWESTMERE\guz

```

```

Start Menu\Programs\Accessories
SQLWESTMERE\guz:Start
Menu\Programs\Accessories
SQLWESTMERE\guz:Start
Start Menu\Programs\Accessories\Accessibility
SQLWESTMERE\guz:Start
Menu\Programs\Accessories\Accessibility
SQLWESTMERE\guz
Start Menu\Programs\Accessories\System Tools
SQLWESTMERE\guz:Start
Menu\Programs\Accessories\System Tools
SQLWESTMERE\guz
Start Menu\Programs\Administrative Tools
SQLWESTMERE\guz:Start
Menu\Programs\Administrative Tools
SQLWESTMERE\guz
Start Menu\Programs\Maintenance
SQLWESTMERE\guz:Start
Menu\Programs\Maintenance
SQLWESTMERE\guz
Start Menu\Programs\Startup
SQLWESTMERE\guz:Start
Menu\Programs\Startup SQLWESTMERE\guz

Start Menu SQLWESTMERE\Admin2:Start Menu
SQLWESTMERE\Admin2
Start Menu\Programs
SQLWESTMERE\Admin2:Start
Menu\Programs
SQLWESTMERE\Admin2
Start Menu\Programs\Accessories
SQLWESTMERE\Admin2:Start
Menu\Programs\Accessories
SQLWESTMERE\Admin2
Start Menu\Programs\Accessories\Accessibility
SQLWESTMERE\Admin2:Start
Menu\Programs\Accessories\Accessibility
SQLWESTMERE\Admin2
Start Menu\Programs\Accessories\System Tools
SQLWESTMERE\Admin2:Start
Menu\Programs\Accessories\System Tools
SQLWESTMERE\Admin2
Start Menu\Programs\Administrative Tools
SQLWESTMERE\Admin2:Start
Menu\Programs\Administrative Tools
SQLWESTMERE\Admin2
Start Menu\Programs\Maintenance
SQLWESTMERE\Admin2:Start
Menu\Programs\Maintenance
SQLWESTMERE\Admin2
Start Menu\Programs\Startup
SQLWESTMERE\Admin2:Start
Menu\Programs\Startup
SQLWESTMERE\Administrator:Start
Menu SQLWESTMERE\Administrator

Start Menu\Programs
SQLWESTMERE\Administrator:Start
Menu\Programs
SQLWESTMERE\Administrator

Start Menu\Programs\Accessories
SQLWESTMERE\Administrator:Start
Menu\Programs\Accessories
SQLWESTMERE\Administrator
Start Menu\Programs\Accessories\Accessibility
SQLWESTMERE\Administrator:Start
Menu\Programs\Accessories\Accessibility
SQLWESTMERE\Administrator

```

Start Menu\Programs\Accessories\System Tools
SQLWESTMERE\Administrator:Start
Menu\Programs\Accessories\System Tools
SQLWESTMERE\Administrator

Start Menu\Programs\Administrative Tools
SQLWESTMERE\Administrator:Start
Menu\Programs\Administrative Tools
SQLWESTMERE\Administrator

Start Menu\Programs\Maintenance
SQLWESTMERE\Administrator:Start
Menu\Programs\Maintenance
SQLWESTMERE\Administrator

Start Menu\Programs\QLogic Management Suite
SQLWESTMERE\Administrator:Start
Menu\Programs\QLogic Management Suite
SQLWESTMERE\Administrator

Start Menu\Programs\Startup
SQLWESTMERE\Administrator:Start
Menu\Programs\Startup
SQLWESTMERE\Administrator

Start Menu\Programs\WinRAR
SQLWESTMERE\Administrator:Start
Menu\Programs\WinRAR
SQLWESTMERE\Administrator

[Startup Programs]

Program	Command	User Name	Location
QLogicSaveSystemInfo	rundll32.exe qlco1006.dll,qlsavesysteminfo	Public	
	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run		

[OLE Registration]

Object	Local Server
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Paintbrush Picture	%systemroot%\system32\mspaint. exe
Package Not Available	
Microsoft PenInputPanel Control Available	Not Available

[Windows Error Reporting]

Time	Type	Details
1/7/2011 9:19 PM	Application Error	Faulting application name: SQLCMD.EXE, version: 2005.90.4035.0, time stamp: 0x492b15af
Faulting module name: kernel32.dll, version: 6.1.7600.16385, time stamp: 0x4a5bdfdf
Exception code: 0xc000005
Fault offset: 0x000000000224a0
Faultin g process id: 0xe40
Faulting application start time: 0x01cbae159cd63d96
Faulting application path: C:\Program Files\Microsoft SQL Server\90\Tools\bin\SQLCMD.EXE&#x 000a;Faulting module path: C:\Windows\system32\kernel32.dll&#x 000a;Report Id: c25ffed0-1aa3-11e0-8e00- 18a9055dc4e6

1/7/2011 9:19 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRASH
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
SQLCMD.EXE
P2:
2005.90.4035.0
P3:
492b15af
P4:
kernel32.dll
P5:
6.1.7600.16385
P6:
4a5bdfdf
P7:
c000005
P8:
000000000224a0
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\Users\Administrator\
AppData\Local\Microsoft\Windows\WER\ReportAr
chive\AppCrash_SQLCMD.EXE_9d43fce1b62f1e63
85f09e167c82126df1d8e_0897579f&#
x000a;
Analysis symbol:

Rechecking for solution:
0
Report Id: c25ffed0-1aa3-
11e0-8e00-
18a9055dc4e6
Report Status:
0

Microsoft SQL Server 2005 Startup Parameters

```
sqlservr.exe -c -x -T661 -T834 -T3502 -T8011  
-T8012 -T8018 -T8033 -T8710 -T8019 -T8744
```

where

```
rem ** -c - Run as console app
rem ** -x - Disable stats
rem ** -T661 - Disable the ghost
record removal process
rem ** -T677 - Disable bunch of
checks in AM. Available in chk/fre, not golden
rem ** -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 ** -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 ** -T9259 - prevent pullup of projects
```

Microsoft SQL Server TCP and Soft Numa Config

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLi
b\Tcp\IPAll]
"TcpPort"=
1433,2001[0x1],2002[0x2],2003[0x4],2004[0x8],
2005[0x10],2006[0x20],2007[0x40],2008[0x80]"
"TcpDynamicPorts"=""
"DisplayName"="Any IP Address"
```

User 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 2005 could use large amounts of physical memory.

Qlogic QMH2562 8 Gb Fibre Channel HBA

All settings left at default

OnBoard Broadcom NIC settings

All default

C.1 Microsoft SQL Server Configuration Parameters

name	minimum	maximum	config_value	run_value
Ad Hoc Distributed Queries	0	1		
0				
affinity I/O mask			-2147483648	
2147483647	0	0		
affinity mask			-2147483648	
2147483647	-2	-2		
affinity64 I/O mask			-2147483648	
2147483647	0	0		
affinity64 mask			-2147483648	
2147483647	65535	65535		

```

Agent XPs 0 1 0
0
allow updates 0 1 0
0
awe enabled 0 1 0
0
blocked process threshold 0 86400
0 0
c2 audit mode 0 1 0
0
clr enabled 0 1 0
0
common criteria compliance enabled 0 1
0 0
cost threshold for parallelism 0 32767
5 5
cross db ownership chaining 0 1
0 0
cursor threshold -1
2147483647 -1 -1
Database Mail XPs 0 1
0 0
default full-text language 0
2147483647 1033 1033
default language 0 9999
0 0
default trace enabled 0 1
0 0
disallow results from triggers 0 1
0 0
fill factor (%) 0 100 0
0
ft crawl bandwidth (max) 0 32767
100 100
ft crawl bandwidth (min) 0 32767
0 0
ft notify bandwidth (max) 0 32767
100 100
ft notify bandwidth (min) 0 32767
0 0
in-doubt xact resolution 0 2
0 0
index create memory (KB) 704
2147483647 0 0
lightweight pooling 0 1
1 1
locks 5000
2147483647 0 0
max degree of parallelism 0 64
48 48
max full-text crawl range 0 256
4 4
max server memory (MB) 16
2147483647 525000 525000
max text repl size (B) 0
2147483647 65536 65536
max worker threads 128 32767
8092 8092media retention
0 365 0 0

min memory per query (KB) 512
min memory per query (KB) 512
2147483647 1024 1024
min server memory (MB) 0
2147483647 0 16
nested triggers 0 1 1
1
network packet size (B) 512 32767
4096 4096
Ole Automation Procedures 0 1
0 0

```

```

Processors 1 3600
2147483647 0 0
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
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 options 0 32767
0 0
Web Assistant Procedures 0 1
0 0
xp_cmdshell 0 1 0
0

Client System Configuration Parameters

```

Client Windows Server 2008 R2 Standard Settings

```

System Information report written at: 05/16/11
14:45:55
System Name: C7KBAY2
[System Summary]

Item Value
OS Name Microsoft Windows Server 2008 R2
Standard
Version 6.1.7600 Build 7600
Other OS Description Not Available
OS Manufacturer Microsoft Corporation

System Name C7KBAY2
System Manufacturer HP
System Model ProLiant BL460c G6

```

```

Processor Intel(R) Xeon(R) CPU E5520
@ 2.27GHz, 2266 Mhz, 4 Core(s), 4 Logical
Processor(s)
Processor Intel(R) Xeon(R) CPU E5520
@ 2.27GHz, 2266 Mhz, 4 Core(s), 4 Logical
Processor(s)
BIOS Version/Date HP I24, 7/25/2009

SMBIOS Version 2.6
Windows Directory C:\Windows
System Directory C:\Windows\system32

Boot Device \Device\HarddiskVolume1
Locale United States
Hardware Abstraction Layer Version
= "6.1.7600.16385"
User Name Not Available
Time Zone Pacific Daylight Time
Installed Physical Memory (RAM) 8.00 GB

Total Physical Memory 7.99 GB
Available Physical Memory 6.97 GB
Total Virtual Memory 16.0 GB
Available Virtual Memory 14.9 GB
Page File Space 7.99 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource Device
I/O Port 0x00000000-0x0000000F Direct
memory access controller
I/O Port 0x00000000-0x0000000F PCI bus

IRQ 20 Intel(R) ICH10 Family USB
Enhanced Host Controller - 3A3A
IRQ 20 Intel(R) ICH10 Family USB
Universal Host Controller - 3A34

I/O Port 0x000003C0-0x000003DF PCI bus

I/O Port 0x000003C0-0x000003DF Standard
VGA Graphics Adapter

I/O Port 0x00000070-0x00000071 System
CMOS/real time clock
I/O Port 0x00000070-0x00000071
Motherboard resources

IRQ 10 IPMI Interface
IRQ 10 Base System Device
IRQ 10 Base System Device

Memory Address 0xE8000000-0xEFFFFFFF
Intel(R) 82801 PCI Bridge - 244E

Memory Address 0xE8000000-0xEFFFFFFF
Standard VGA Graphics Adapter

IRQ 22 Intel(R) ICH10 Family USB
Universal Host Controller - 3A36
IRQ 22 Standard Universal PCI to USB Host
Controller

IRQ 23 Intel(R) ICH10 Family USB
Universal Host Controller - 3A35
IRQ 23 Intel(R) ICH10 Family USB
Universal Host Controller - 3A39

```

Memory Address 0xFED00000-0xFED003FF High precision event timer
Memory Address 0xFED00000-0xFED003FF PCI bus
Memory Address 0xFED00000-0xFED003FF PCI bus
Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF Standard VGA Graphics Adapter
I/O Port 0x000003B0-0x000003BB PCI bus
I/O Port 0x000003B0-0x000003BB Standard VGA Graphics Adapter
I/O Port 0x00001000-0x00004FFF PCI bus
I/O Port 0x00001000-0x00004FFF Intel(R) ICH10 Family USB Universal Host Controller - 3A34

[DMA]

Resource	Device	Status
Channel 7	Direct memory access controller	OK

[Forced Hardware]

Device	PNP Device ID

[I/O]

Resource	Device	Status
0x00000040-0x00000043	System timer	OK
0x00001020-0x0000103F	Intel(R) ICH10 Family USB Universal Host Controller - 3A35	OK
0x00000000-0x0000000F	Direct memory access controller	OK
0x00000000-0x0000000F	PCI bus	OK
0x00000080-0x0000008F	Direct memory access controller	OK
0x000000C0-0x000000DF	Direct memory access controller	OK
0x00001040-0x0000105F	Intel(R) ICH10 Family USB Universal Host Controller - 3A36	OK
0x00000060-0x00000060	Standard PS/2 Keyboard	OK
0x00000064-0x00000064	Standard PS/2 Keyboard	OK
0x000002F8-0x000002FF	Communications Port (COM1)	OK
0x00000061-0x00000061	System speaker	OK
0x00003800-0x0000381F	Standard Universal PCI to USB Host Controller	OK
0x00001060-0x0000107F	Intel(R) ICH10 Family USB Universal Host Controller - 3A39	OK
0x00001000-0x00004FFF	PCI bus	OK
0x00001000-0x00004FFF	Intel(R) ICH10 Family USB Universal Host Controller - 3A34	OK
0x000003E0-0x00000CF7	PCI bus	OK
0x00000D00-0x00000FFF	PCI bus	OK

0x000003B0-0x000003BB	PCI bus	OK
0x000003B0-0x000003BB	Standard VGA Graphics Adapter	OK
0x000003C0-0x000003DF	PCI bus	OK
0x000003C0-0x000003DF	Standard VGA Graphics Adapter	OK
0x0000002E-0x0000002F	Extended IO Bus	OK
0x00000620-0x0000065F	Extended IO Bus	OK
0x00000680-0x0000069F	Extended IO Bus	OK
0x00000600-0x0000061F	Extended IO Bus	OK
0x00000660-0x0000067F	Extended IO Bus	OK
0x00000300-0x0000031F	Extended IO Bus	OK
0x00000070-0x00000071	System CMOS/real time clock	OK
0x00000070-0x00000071	Motherboard resources	OK
0x00000408-0x0000040F	Motherboard resources	OK
0x000004D0-0x000004D1	Motherboard resources	OK
0x00000020-0x0000003F	Motherboard resources	OK
0x000000A0-0x000000BF	Motherboard resources	OK
0x00000090-0x0000009F	Motherboard resources	OK
0x00000050-0x00000053	Motherboard resources	OK
0x00000700-0x0000071F	Motherboard resources	OK
0x00000880-0x000008FF	Motherboard resources	OK
0x00000900-0x0000097F	Motherboard resources	OK
0x00000010-0x0000001F	Motherboard resources	OK
0x00000C80-0x00000C83	Motherboard resources	OK
0x00000CD4-0x00000CD7	Motherboard resources	OK
0x00000F50-0x00000F58	Motherboard resources	OK
0x000000F0-0x000000F0	Motherboard resources	OK
0x00000CA0-0x00000CA1	Motherboard resources	OK
0x00000CA4-0x00000CA5	Motherboard resources	OK
0x000003F8-0x000003FF	Motherboard resources	OK
0x00004000-0x00004FFF	Intel(R) ICH10 Family PCI Express Root Port 1 - 3A40	OK
0x00002800-0x000028FF	Base System Device	OK
0x00002000-0x00003FFF	Intel(R) 82801 PCI Bridge - 244E0K	OK
0x00003400-0x000034FF	Base System Device	OK
0x00000CA2-0x00000CA3	Microsoft Generic IPMI Compliant Device	OK
0x00003000-0x000030FF	Standard VGA Graphics Adapter	OK

[IRQs]

Resource	Device	Status
IRQ 0	System timer	OK

IRQ 4294967294	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 1 - 3A35	OK
IRQ 23	Intel(R) ICH10 Family USB Universal Host Controller - 3A39	OK
IRQ 4294967287	Smart Array Controller (Media Driver)	OK
IRQ 4294967293	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 2 - 3409	OK
IRQ 22	Intel(R) ICH10 Family USB Universal Host Controller - 3A36	OK
IRQ 22	Standard Universal PCI to USB Host Controller	OK
IRQ 1	Standard PS/2 Keyboard	OK
IRQ 3	Communications Port (COM1)	OK
IRQ 4294967292	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 3 - 340A	OK
IRQ 10	IPMI Interface	OK
IRQ 10	Base System Device	OK
IRQ 10	Base System Device	OK
IRQ 4294967291	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 7 - 340E	OK
IRQ 20	Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3A	OK
IRQ 20	Intel(R) ICH10 Family USB Universal Host Controller - 3A34	OK
IRQ 12	PS/2 Compatible Mouse	OK
IRQ 28	Broadcom BCM57711E NetXtreme II 10 GigE	OK
IRQ 4294967290	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 8 - 340F	OK
IRQ 16	Intel(R) ICH10 Family PCI Express Root Port 1 - 3A40	OK
IRQ 81	Microsoft ACPI-Compliant System	OK
IRQ 82	Microsoft ACPI-Compliant System	OK
IRQ 83	Microsoft ACPI-Compliant System	OK
IRQ 84	Microsoft ACPI-Compliant System	OK
IRQ 85	Microsoft ACPI-Compliant System	OK
IRQ 86	Microsoft ACPI-Compliant System	OK
IRQ 87	Microsoft ACPI-Compliant System	OK
IRQ 88	Microsoft ACPI-Compliant System	OK
IRQ 89	Microsoft ACPI-Compliant System	OK
IRQ 90	Microsoft ACPI-Compliant System	OK
IRQ 91	Microsoft ACPI-Compliant System	OK
IRQ 92	Microsoft ACPI-Compliant System	OK
IRQ 93	Microsoft ACPI-Compliant System	OK

IRQ 94 Microsoft ACPI-Compliant System OK
 IRQ 95 Microsoft ACPI-Compliant System OK
 IRQ 96 Microsoft ACPI-Compliant System OK
 IRQ 97 Microsoft ACPI-Compliant System OK
 IRQ 98 Microsoft ACPI-Compliant System OK
 IRQ 99 Microsoft ACPI-Compliant System OK
 IRQ 100 Microsoft ACPI-Compliant System OK
 IRQ 101 Microsoft ACPI-Compliant System OK
 IRQ 102 Microsoft ACPI-Compliant System OK
 IRQ 103 Microsoft ACPI-Compliant System OK
 IRQ 104 Microsoft ACPI-Compliant System OK
 IRQ 105 Microsoft ACPI-Compliant System OK
 IRQ 106 Microsoft ACPI-Compliant System OK
 IRQ 107 Microsoft ACPI-Compliant System OK
 IRQ 108 Microsoft ACPI-Compliant System OK
 IRQ 109 Microsoft ACPI-Compliant System OK
 IRQ 110 Microsoft ACPI-Compliant System OK
 IRQ 111 Microsoft ACPI-Compliant System OK
 IRQ 112 Microsoft ACPI-Compliant System OK
 IRQ 113 Microsoft ACPI-Compliant System OK
 IRQ 114 Microsoft ACPI-Compliant System OK
 IRQ 115 Microsoft ACPI-Compliant System OK
 IRQ 116 Microsoft ACPI-Compliant System OK
 IRQ 117 Microsoft ACPI-Compliant System OK
 IRQ 118 Microsoft ACPI-Compliant System OK
 IRQ 119 Microsoft ACPI-Compliant System OK
 IRQ 120 Microsoft ACPI-Compliant System OK
 IRQ 121 Microsoft ACPI-Compliant System OK
 IRQ 122 Microsoft ACPI-Compliant System OK
 IRQ 123 Microsoft ACPI-Compliant System OK
 IRQ 124 Microsoft ACPI-Compliant System OK
 IRQ 125 Microsoft ACPI-Compliant System OK
 IRQ 126 Microsoft ACPI-Compliant System OK
 IRQ 127 Microsoft ACPI-Compliant System OK
 IRQ 128 Microsoft ACPI-Compliant System OK
 IRQ 129 Microsoft ACPI-Compliant System OK
 IRQ 130 Microsoft ACPI-Compliant System OK

IRQ 131 Microsoft ACPI-Compliant System OK
 IRQ 132 Microsoft ACPI-Compliant System OK
 IRQ 133 Microsoft ACPI-Compliant System OK
 IRQ 134 Microsoft ACPI-Compliant System OK
 IRQ 135 Microsoft ACPI-Compliant System OK
 IRQ 136 Microsoft ACPI-Compliant System OK
 IRQ 137 Microsoft ACPI-Compliant System OK
 IRQ 138 Microsoft ACPI-Compliant System OK
 IRQ 139 Microsoft ACPI-Compliant System OK
 IRQ 140 Microsoft ACPI-Compliant System OK
 IRQ 141 Microsoft ACPI-Compliant System OK
 IRQ 142 Microsoft ACPI-Compliant System OK
 IRQ 143 Microsoft ACPI-Compliant System OK
 IRQ 144 Microsoft ACPI-Compliant System OK
 IRQ 145 Microsoft ACPI-Compliant System OK
 IRQ 146 Microsoft ACPI-Compliant System OK
 IRQ 147 Microsoft ACPI-Compliant System OK
 IRQ 148 Microsoft ACPI-Compliant System OK
 IRQ 149 Microsoft ACPI-Compliant System OK
 IRQ 150 Microsoft ACPI-Compliant System OK
 IRQ 151 Microsoft ACPI-Compliant System OK
 IRQ 152 Microsoft ACPI-Compliant System OK
 IRQ 153 Microsoft ACPI-Compliant System OK
 IRQ 154 Microsoft ACPI-Compliant System OK
 IRQ 155 Microsoft ACPI-Compliant System OK
 IRQ 156 Microsoft ACPI-Compliant System OK
 IRQ 157 Microsoft ACPI-Compliant System OK
 IRQ 158 Microsoft ACPI-Compliant System OK
 IRQ 159 Microsoft ACPI-Compliant System OK
 IRQ 160 Microsoft ACPI-Compliant System OK
 IRQ 161 Microsoft ACPI-Compliant System OK
 IRQ 162 Microsoft ACPI-Compliant System OK
 IRQ 163 Microsoft ACPI-Compliant System OK
 IRQ 164 Microsoft ACPI-Compliant System OK
 IRQ 165 Microsoft ACPI-Compliant System OK
 IRQ 166 Microsoft ACPI-Compliant System OK
 IRQ 167 Microsoft ACPI-Compliant System OK
 IRQ 168 Microsoft ACPI-Compliant System OK

IRQ 169 Microsoft ACPI-Compliant System OK
 IRQ 170 Microsoft ACPI-Compliant System OK
 IRQ 171 Microsoft ACPI-Compliant System OK
 IRQ 172 Microsoft ACPI-Compliant System OK
 IRQ 173 Microsoft ACPI-Compliant System OK
 IRQ 174 Microsoft ACPI-Compliant System OK
 IRQ 175 Microsoft ACPI-Compliant System OK
 IRQ 176 Microsoft ACPI-Compliant System OK
 IRQ 177 Microsoft ACPI-Compliant System OK
 IRQ 178 Microsoft ACPI-Compliant System OK
 IRQ 179 Microsoft ACPI-Compliant System OK
 IRQ 180 Microsoft ACPI-Compliant System OK
 IRQ 181 Microsoft ACPI-Compliant System OK
 IRQ 182 Microsoft ACPI-Compliant System OK
 IRQ 183 Microsoft ACPI-Compliant System OK
 IRQ 184 Microsoft ACPI-Compliant System OK
 IRQ 185 Microsoft ACPI-Compliant System OK
 IRQ 186 Microsoft ACPI-Compliant System OK
 IRQ 187 Microsoft ACPI-Compliant System OK
 IRQ 188 Microsoft ACPI-Compliant System OK
 IRQ 189 Microsoft ACPI-Compliant System OK
 IRQ 190 Microsoft ACPI-Compliant System OK
 IRQ 47 Broadcom BCM57711E NetXtreme II 10 GigE OK
 IRQ 4294967289 Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 9 - 3410 OK
 IRQ 4294967288 Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 10 - 3411 OK
 [Memory]

Resource	Device	Status
0xF3800000-0xFB7FFFFF	Intel(R) 5520/5500/X58 I/O Hub PCI Express Root Port 1 - 3408	OK
0xFED00000-0xFED003FF	High precision event timer	OK
0xFED00000-0xFED003FF	PCI bus	OK
0xFED00000-0xFED003FF	PCI bus	OK
0xFBC00000-0xFBFFFFFF	Smart Array Controller (Media Driver)	OK
0xFBBF0000-0xFBFF0FFF	Smart Array Controller (Media Driver)	OK
0xE7000000-0xFBFFFFFF	PCI bus	OK
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	Standard VGA Graphics Adapter	OK

```

0xF36F0000-0xF36F00FF IPMI Interface
                        OK
0xF35F0000-0xF35F03FF Intel(R) ICH10 Family
USB Enhanced Host Controller - 3A3A OK

0xE0000000-0xE3FFFFFF Motherboard
resources OK
0xFE000000-0xFE8FFFFFF Motherboard
resources OK
0xE7FFE000-0xE7FFFFFF Motherboard
resources OK
0xFB000000-0xFB7FFFFFF Broadcom BCM57711E
NetXtreme II 10 GiGE OK
0xFA800000-0xFAFFFFFF Broadcom BCM57711E
NetXtreme II 10 GiGE OK
0xFBB00000-0xFBFFFFFF Intel(R) ICH10 Family
PCI Express Root Port 1 - 3A40 OK

0xFA000000-0xFA7FFFFFF Broadcom BCM57711E
NetXtreme II 10 GiGE OK
0xF9800000-0xF9FFFFFF Broadcom BCM57711E
NetXtreme II 10 GiGE OK
0xF37E0000-0xF37E01FF Base System Device
                        OK
0xF3600000-0xF37FFFFFF Intel(R) 82801 PCI
Bridge - 244EOK
0xE8000000-0xEFFFFFF Intel(R) 82801 PCI
Bridge - 244EOK
0xE8000000-0xEFFFFFF Standard VGA
Graphics Adapter OK
0xF37D0000-0xF37D07FF Base System Device
                        OK
0xF37C0000-0xF37C3FFF Base System Device
                        OK
0xF3700000-0xF377FFFF Base System Device
                        OK
0xF37F0000-0xF377FFFF Standard VGA
Graphics Adapter OK

```

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer File	Description Version	Status Size
c:\windows\system32\msg711.acm			
M	C:\Windows\system32\MSG711.AC	Microsoft Corporation	14.50 KB (14,848 bytes)
[Video Codecs]			
CODEC Manufacturer Description Status			
File Version Size			
Creation Date			
c:\windows\system32\tsbyuv.dll			
M	C:\Windows\system32\TSBYUV.DLL	Microsoft Corporation	14.50 KB (14,848 bytes)
[Video Codecs]			
CODEC Manufacturer Description Status			
File Version Size			
Creation Date			
c:\windows\system32\iyuv_32.dll			
L	C:\Windows\system32\IYUV_32.DL	Microsoft Corporation	53.00 KB (54,272 bytes)
c:\windows\system32\msyuv.dll			
L	C:\Windows\system32\MSYUV.DLL	Microsoft Corporation	24.50 KB (25,088 bytes)
c:\windows\system32\msrle32.dll			
L	C:\Windows\system32\MSRLE32.DL	Microsoft Corporation	16.00 KB (16,384 bytes)
c:\windows\system32\msvidc32.dll			
LL	C:\Windows\system32\MSVIDC32.D	Microsoft Corporation	38.00 KB (38,912 bytes)

```

c:\windows\system32\msg711.acm
Microsoft Corporation
OK
C:\Windows\system32\MSG711.AC
6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 5:18 PM

[Video Codecs]

CODEC Manufacturer Description Status
File Version Size
Creation Date
c:\windows\system32\tsbyuv.dll
Microsoft Corporation
OK
C:\Windows\system32\TSBYUV.DLL
6.1.7600.16490 14.50 KB
(14,848 bytes) 9/15/2010 10:46 AM

c:\windows\system32\iyuv_32.dll
Microsoft Corporation
OK
C:\Windows\system32\IYUV_32.DL
6.1.7600.16490 53.00 KB
(54,272 bytes) 9/15/2010 10:46 AM

c:\windows\system32\msyuv.dll
Microsoft Corporation
OK
C:\Windows\system32\MSYUV.DLL
6.1.7600.16490 24.50 KB
(25,088 bytes) 9/15/2010 10:46 AM

c:\windows\system32\msrle32.dll
Microsoft Corporation
OK
C:\Windows\system32\MSRLE32.DL
6.1.7600.16490 16.00 KB
(16,384 bytes) 9/15/2010 10:46 AM

c:\windows\system32\msvidc32.dll
Microsoft Corporation
OK
C:\Windows\system32\MSVIDC32.D
6.1.7600.16490 38.00 KB
(38,912 bytes) 9/15/2010 10:46 AM

[CD-ROM]

Item Value

[Sound Device]

Item Value

[Display]

Item Value
Name Standard VGA Graphics Adapter

PNP Device ID PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_02\4&10AF73B4&0&18F0

Adapter TypeNot Available, (Standard display
types) compatible
Adapter Description Standard VGA
Graphics Adapter
Adapter RAM Not Available
Installed Drivers Not Available
Driver Version 6.1.7600.16385

INF File display.inf (vga section)
Color Planes Not Available

```

```

Resolution Not Available Not Available
Bits/Pixel Not Available
Memory Address 0xE8000000-
0xEFFFFFFF
I/O Port 0x00003000-0x000030FF
Memory Address 0xF37F0000-
0xF37FFFFFF
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF

Driver
c:\windows\system32\drivers\vgapn
p.sys (6.1.7600.16385, 28.50 KB (29,184 bytes),
7/13/2009 4:38 PM)

[Infrared]

Item Value

[Input]

Item Value

[Keyboard]

Item Value
Description USB Input Device
Name Enhanced (101- or 102-key)

Layout 00000409
PNP Device ID
USB\VID_03F0&PID_1027&MI_00\7
&EEB5304&0&0000
Number of Function Keys 12
Driver
c:\windows\system32\drivers\hidus
b.sys (6.1.7600.16385, 29.50 KB (30,208 bytes),
7/13/2009 5:06 PM)

[Pointing Device]

Item Value
Hardware Type USB Input Device

Number of Buttons 0
Status OK
PNP Device ID
USB\VID_03F0&PID_1027&MI_01\7
&EEB5304&0&0001
Power Management Supported No

Double Click Threshold Not Available
Handedness Not Available
Driver
c:\windows\system32\drivers\hidus
b.sys (6.1.7600.16385, 29.50 KB (30,208 bytes),
7/13/2009 5:06 PM)

[Modem]

Item Value

[Network]

Item Value
Name [00000000] WAN Miniport (SSTP)

Adapter TypeNot Available
Product TypeWAN Miniport (SSTP)
Installed Yes

```

PNP Device ID
ROOT\MS_SSTPMINIPOINT\0000

Last Reset 5/11/2011 11:35 AM
Index 0
Service Name RasSstp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\rasstp.sys (6.1.7600.16385, 82.00 KB (83,968 bytes), 7/13/2009 5:10 PM)

Name [00000001] WAN Miniport (IKeV2)

Adapter TypeNot Available
Product TypeWAN Miniport (IKeV2)
Installed Yes
PNP Device ID
ROOT\MS_AGILEVPMINIPOINT\0000

Last Reset 5/11/2011 11:35 AM
Index 1
Service Name RasAgileVpn
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\agilev.pn.sys (6.1.7600.16385, 59.00 KB (60,416 bytes), 7/13/2009 5:10 PM)

Name [00000002] WAN Miniport (L2TP)

Adapter TypeNot Available
Product TypeWAN Miniport (L2TP)
Installed Yes
PNP Device ID
ROOT\MS_L2TPMINIPOINT\0000

Last Reset 5/11/2011 11:35 AM
Index 2
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\rasl2tp.sys (6.1.7600.16385, 127.00 KB (130,048 bytes), 7/13/2009 5:10 PM)

Name [00000003] WAN Miniport (PPTP)

Adapter TypeNot Available
Product TypeWAN Miniport (PPTP)
Installed Yes
PNP Device ID
ROOT\MS_PPTPMINIPOINT\0000

Last Reset 5/11/2011 11:35 AM
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 Not Available
Driver
c:\windows\system32\drivers\rasppp.sys (6.1.7600.16385, 109.00 KB (111,616 bytes), 7/13/2009 5:10 PM)

Name [00000004] WAN Miniport (PPPOE)

Adapter TypeNot Available
Product TypeWAN Miniport (PPPOE)
Installed Yes
PNP Device ID
ROOT\MS_PPPOEMINIPOINT\0000

Last Reset 5/11/2011 11:35 AM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\rasppoe.sys (6.1.7600.16385, 90.50 KB (92,672 bytes), 7/13/2009 5:10 PM)

Name [00000005] WAN Miniport (IPv6)

Adapter TypeNot Available
Product TypeWAN Miniport (IPv6)
Installed Yes
PNP Device ID
ROOT\MS_NDISWANIPV6\0000

Last Reset 5/11/2011 11:35 AM
Index 5
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\ndiswan.sys (6.1.7600.16385, 160.50 KB (164,352 bytes), 7/13/2009 5:10 PM)

Name [00000006] WAN Miniport (Network Monitor)
Adapter TypeNot Available
Product TypeWAN Miniport (Network Monitor)
Installed Yes
PNP Device ID
ROOT\MS_NDISWANBH\0000

Last Reset 5/11/2011 11:35 AM
Index 6
Service Name NdisWan

IP Address Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\ndiswan.sys (6.1.7600.16385, 160.50 KB (164,352 bytes), 7/13/2009 5:10 PM)

Name [00000007] Broadcom BCM57711E NetXtreme II 10 GigE (NDIS VBD Client)
Adapter TypeEthernet 802.3
Product TypeBroadcom BCM57711E NetXtreme II 10 GigE (NDIS VBD Client)
Installed Yes
PNP Device ID
EBDRV\L2ND&PCI_165014E4&SUBSYS_7058103C&REV_00\5&2827D6F3&0&20050200

Last Reset 5/11/2011 11:35 AM
Index 7
Service Name l2nd
IP Address 192.168.1.2, fe80::5457:cd89:37b4:76a0
IP Subnet 255.255.0.0, 64
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:26:55:7D:4C:68
Driver
c:\windows\system32\drivers\bxdn60a.sys (4.8.4.0, 70.00 KB (71,680 bytes), 6/10/2009 1:34 PM)

Name [00000008] Microsoft ISATAP
Adapter
Adapter TypeTunnel
Product TypeMicrosoft ISATAP Adapter
Installed Yes
PNP Device ID
ROOT*ISATAP\0000

Last Reset 5/11/2011 11:35 AM
Index 8
Service Name tunnel
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver
c:\windows\system32\drivers\tunnel.sys (6.1.7600.16385, 122.50 KB (125,440 bytes), 7/13/2009 5:09 PM)

Name [00000009] WAN Miniport (IP)

Adapter TypeNot Available
Product TypeWAN Miniport (IP)
Installed Yes
PNP Device ID
ROOT\MS_NDISWANIP\0000

Last Reset 5/11/2011 11:35 AM
Index 9
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available

Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver
 c:\windows\system32\drivers\ndisw
 an.sys (6.1.7600.16385, 160.50 KB (164,352
 bytes), 7/13/2009 5:10 PM)

 Name [00000010] Microsoft 6to4 Adapter

 Adapter TypeTunnel
 Product TypeMicrosoft 6to4 Adapter
 Installed Yes
 PNP Device ID ROOT*6TO4MP\0000

 Last Reset 5/11/2011 11:35 AM
 Index 10
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver
 c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

 Name [00000011] RAS Async Adapter

 Adapter TypeWide Area Network (WAN)
 Product TypeRAS Async Adapter
 Installed Yes
 PNP Device ID SW\{EEAB7790-C514-
 11D1-B42B-00805FC1270E}\ASYNCMAC
 Last Reset 5/11/2011 11:35 AM
 Index 11
 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 20:41:53:59:4E:FF
 Driver
 c:\windows\system32\drivers\async
 mac.sys (6.1.7600.16385, 22.50 KB (23,040
 bytes), 7/13/2009 5:10 PM)

 Name [00000012] Broadcom BCM57711E
 NetXtreme II 10 GigE (NDIS VBD Client)
 Adapter TypeEthernet 802.3
 Product TypeBroadcom BCM57711E NetXtreme
 II 10 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID
 EBDRV\L2ND&PCI_165014E4&SUBS
 YS_7058103C&REV_00\5&F8EAF49&0&20050200

 Last Reset 5/11/2011 11:35 AM
 Index 12
 Service Name l2nd
 IP Address 15.1.115.2,
 fe80::fdca:55ee:86a:e84a
 IP Subnet 255.255.0.0, 64
 Default IP Gateway Not Available
 DHCP Enabled No

DHCP Server Not Available Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:26:55:7D:4C:6C
 Driver
 c:\windows\system32\drivers\bxnd
 60a.sys (4.8.4.0, 70.00 KB (71,680 bytes),
 6/10/2009 1:34 PM)

 Name [00000013] Microsoft ISATAP
 Adapter
 Adapter TypeTunnel
 Product TypeMicrosoft ISATAP Adapter
 Installed Yes
 PNP Device ID ROOT*ISATAP\0001

 Last Reset 5/11/2011 11:35 AM
 Index 13
 Service Name tunnel
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available
 Driver
 c:\windows\system32\drivers\tunne
 l.sys (6.1.7600.16385, 122.50 KB (125,440
 bytes), 7/13/2009 5:09 PM)

 [Protocol]

 Item Value
 Name MSAFD Tcpip [TCP/IP]
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 16 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No

 Supports Multicasting No

 Name MSAFD Tcpip [UDP/IP]
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.99 KB (65,527
 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 MSAFD Tcpip [TCP/IPv6]
 Connectionless Service No

Guarantees Delivery Yes
 Maximum Address Size 28 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 28 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/IPv6]
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 28 bytes
 Maximum Message Size 63.99 KB (65,527
 bytes)
 Message Oriented Yes
 Minimum Address Size 28 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 TCPv6 Service Provider
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 28 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 28 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No

 Supports Multicasting No

 Name RSVP TCP Service Provider
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 16 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No

 Supports Multicasting No

Name RSVP UDPv6 Service Provider

Connectionless Service Yes

Guarantees Delivery No

Guarantees Sequencing No

Maximum Address Size 28 bytes

Maximum Message Size 63.99 KB (65,527 bytes)

Message Oriented Yes

Minimum Address Size 28 bytes

Pseudo Stream Oriented No

Supports Broadcasting Yes

Supports Connect Data No

Supports Disconnect Data No

Supports Encryption Yes

Supports Expedited Data No

Supports Graceful Closing No

Supports Guaranteed Bandwidth No

Supports Multicasting Yes

Name RSVP UDP Service Provider

Connectionless Service Yes

Guarantees Delivery No

Guarantees Sequencing No

Maximum Address Size 16 bytes

Maximum Message Size 63.99 KB (65,527 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

[WinSock]

Item Value

File c:\windows\syswow64\wssock32.dll

Size 15.00 KB (15,360 bytes)

Version 6.1.7600.16385

File c:\windows\system32\wssock32.dll

Size 18.00 KB (18,432 bytes)

Version 6.1.7600.16385

[Ports]

[Serial]

Item Value

Name Communications Port (COM1)

Status OK

PNP Device ID ACPI\PNP0501\0

Maximum Input Buffer Size0

Maximum Output Buffer Size No

Settable Baud Rate Yes

Settable Data Bits Yes

Settable Flow Control Yes

Settable Parity Yes

Settable Parity Check Yes

Settable Stop Bits Yes

Supports RLLSD Yes

Supports 16 Bit Mode No

Supports Special Characters No

Baud Rate 9600

Bits/Byte 8

Stop Bits 1

Parity None

Busy No

Abort Read/Write on Error No

Binary Mode Enabled Yes

Continue XMit on XOff No

CTS Outflow Control No

Discard NULL Bytes No

DSR Outflow Control 0

DSR Sensitivity 0

DTR Flow Control Type Enable

EOF Character 0

Error Replace Character 0

Error Replacement Enabled No

Event Character 0

Parity Check Enabled No

RTS Flow Control Type Enable

XOff Character 19

XOffXMit Threshold 512

XOn Character 17

XOnXMit Threshold 2048

XOnXOff InFlow Control 0

XOnXOff OutFlow Control 0

IRQ Channel IRQ 3

I/O Port 0x000002F8-0x000002FF

Driver c:\windows\system32\drivers\serial.sys (6.1.7600.16385, 92.00 KB (94,208 bytes), 7/13/2009 5:00 PM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value

Drive C:

Description Local Fixed Disk

Compressed No

File System NTFS

Size 136.60 GB (146,671,661,056 bytes)

Free Space 110.90 GB (119,082,024,960 bytes)

Volume Name

Volume Serial Number 2891ED8E

[Disks]

Item Value

Description Disk drive

Manufacturer (Standard disk drives)

Model HP LOGICAL VOLUME SCSI Disk Device

Bytes/Sector 512

Media Loaded Yes

Media Type Fixed hard disk

Partitions 2

SCSI Bus 0

SCSI Logical Unit 0

SCSI Port 0

SCSI Target ID 4

Sectors/Track 32

Size 136.70 GB (146,778,685,440 bytes)

Total Cylinders 35,132

Total Sectors 286,677,120

Total Tracks 8,958,660

Tracks/Cylinder 255

Partition Disk #0, Partition #0

Partition Size 100.00 MB (104,857,600 bytes)

Partition Starting Offset 1,048,576 bytes

Partition Disk #0, Partition #1

Partition Size 136.60 GB (146,671,665,152 bytes)

Partition Starting Offset 105,906,176 bytes

[SCSI]

Item Value

Name Smart Array Controller (Media Driver)

Manufacturer Hewlett-Packard Company

Status OK

PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3245103C&REV_01\4&294C1E27&0&00E0

Memory Address 0xFBC00000-0xFBFFFFFF

Memory Address 0xFBFBF000-0xFBFBF0FF

IRQ Channel IRQ 4294967287

Driver c:\windows\system32\drivers\hpsa.md.sys (6.12.4.64, 76.06 KB (77,888 bytes), 7/13/2009 2:59 PM)

[IDE]

Item Value

[Printing]

Name Driver Port Name Server Name

Microsoft XPS Document Writer

Microsoft XPS Document Writer

XPSPort: Not Available

[Problem Devices]

Device PNP Device ID Error Code

Standard PS/2 Keyboard

ACPI\PNP0303\4&1CFF2C4C&0

This device is not present, is not working properly, or does not have all its drivers installed.

IPMI Interface

PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_00\4&10AF73B4&0&26F0

The drivers for this device are not installed.

PS/2 Compatible Mouse

ACPI\PNP0F13\4&1CFF2C4C&0

This device is not present, is not working properly, or does not have all its drivers installed.

Base System Device

PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_03\4&10AF73B4&0&20F0

The drivers for this device are not installed.

Base System Device
 PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_03\4&10AF73B4&0&22F0
 The drivers for this device are not installed.

[USB]

Device PNP Device ID
 Intel(R) ICH10 Family USB Universal Host Controller - 3A35
 PCI\VEN_8086&DEV_3A35&SUBSYS_330D103C&REV_00\3&267A616A&0&E9

Intel(R) ICH10 Family USB Universal Host Controller - 3A36
 PCI\VEN_8086&DEV_3A36&SUBSYS_330D103C&REV_00\3&267A616A&0&EA

Standard Universal PCI to USB Host Controller
 PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_00\4&10AF73B4&0&24F0

Intel(R) ICH10 Family USB Universal Host Controller - 3A39
 PCI\VEN_8086&DEV_3A39&SUBSYS_330D103C&REV_00\3&267A616A&0&EB

Intel(R) ICH10 Family USB Enhanced Host Controller - 3A3A
 PCI\VEN_8086&DEV_3A3A&SUBSYS_330D103C&REV_00\3&267A616A&0&EF

Intel(R) ICH10 Family USB Universal Host Controller - 3A34
 PCI\VEN_8086&DEV_3A34&SUBSYS_330D103C&REV_00\3&267A616A&0&E8

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State
Pause	Accept Stop					
1394ohci Controller	1394 OHCI Compliant Host					
ohci.sys	c:\windows\system32\drivers\1394ohci.sys		Kernel Driver	No	Manual	Stopped OK Normal
acpi	Microsoft ACPI Driver					
acpi.sys	c:\windows\system32\drivers\acpi.sys		Kernel Driver	Yes	Boot	Running OK Critical
acpipmi	ACPI Power Meter Driver					
mi.sys	c:\windows\system32\drivers\acpipmi.sys		Kernel Driver	Yes	Manual	Running OK Normal
adp94xx	adp94xx					
4xx.sys	c:\windows\system32\drivers\adp94xx.sys		Kernel Driver	No	Manual	Stopped OK Normal
adpahci	adpahci					
hci.sys	c:\windows\system32\drivers\adpahci.sys		Kernel Driver	No	Manual	Stopped OK Normal

adpu320	adpu320					
320.sys	c:\windows\system32\drivers\adpu320.sys		Kernel Driver	No	Manual	Stopped OK Normal
afd	Microsoft Ancillary Function Driver for Winsock					
afd.sys	c:\windows\system32\drivers\afd.sys		Kernel Driver	Yes	System	Running OK Normal
agp440	Intel AGP Bus Filter					
40.sys	c:\windows\system32\drivers\agp440.sys		Kernel Driver	No	Manual	Stopped OK Normal
aliide	aliide					
aliide.sys	c:\windows\system32\drivers\aliide.sys		Kernel Driver	No	Manual	Stopped OK Critical
amdide	amdide					
de.sys	c:\windows\system32\drivers\amdide.sys		Kernel Driver	No	Manual	Stopped OK Critical
amdk8	AMD K8 Processor Driver					
8.sys	c:\windows\system32\drivers\amdk8.sys		Kernel Driver	No	Manual	Stopped OK Normal
amdppm	AMD Processor Driver					
pm.sys	c:\windows\system32\drivers\amdppm.sys		Kernel Driver	No	Manual	Stopped OK Normal
amdsata	amdsata					
ata.sys	c:\windows\system32\drivers\amdsata.sys		Kernel Driver	No	Manual	Stopped OK Normal
amdsbs	amdsbs					
bs.sys	c:\windows\system32\drivers\amdsbs.sys		Kernel Driver	No	Manual	Stopped OK Normal
amdxata	amdxata					
ata.sys	c:\windows\system32\drivers\amdxata.sys		Kernel Driver	Yes	Boot	Running OK Normal
appid	AppID Driver					
.sys	c:\windows\system32\drivers\appid.sys		Kernel Driver	No	Manual	Stopped OK Normal
arc	arc					
s	c:\windows\system32\drivers\arc.sys		Kernel Driver	No	Manual	Stopped OK Normal
arcsas	arcsas					
s.sys	c:\windows\system32\drivers\arcsas.sys		Kernel Driver	No	Manual	Stopped OK Normal
asyncmac	RAS Asynchronous Media Driver					
mac.sys	c:\windows\system32\drivers\asyncmac.sys		Kernel Driver	Yes	Manual	Running OK Normal
ataapi	ataapi					
sys	c:\windows\system32\drivers\ataapi.sys		Kernel Driver	No	Manual	Stopped OK Critical

b06bdrv	Broadcom NetXtreme II VBD					
a.sys	c:\windows\system32\drivers\b06bdrv.sys		Kernel Driver	No	Manual	Stopped OK Normal
b57nd60a	Broadcom NetXtreme Gigabit Ethernet - NDIS 6.0					
d60a.sys	c:\windows\system32\drivers\b57nd60a.sys		Kernel Driver	No	Manual	Stopped OK Normal
beep	Beep					
sys	c:\windows\system32\drivers\beep.sys		Kernel Driver	No	Manual	Stopped OK Normal
blbdrive	blbdrive					
ve.sys	c:\windows\system32\drivers\blbdrive.sys		Kernel Driver	Yes	System	Running OK Normal
browser	Browser Support Driver					
er.sys	c:\windows\system32\drivers\browser.sys		File System Driver	Yes	Manual	Running OK Normal
brfiltlo	Brother USB Mass-Storage Lower Filter Driver					
o.sys	c:\windows\system32\drivers\brfiltlo.sys		Kernel Driver	No	Manual	Stopped OK Normal
brfiltup	Brother USB Mass-Storage Upper Filter Driver					
p.sys	c:\windows\system32\drivers\brfiltup.sys		Kernel Driver	No	Manual	Stopped OK Normal
brserid	Brother MFC Serial Port Interface Driver (WDM)					
d.sys	c:\windows\system32\drivers\brserid.sys		Kernel Driver	No	Manual	Stopped OK Normal
brserwdm	Brother WDM Serial driver					
wdm.sys	c:\windows\system32\drivers\brserwdm.sys		Kernel Driver	No	Manual	Stopped OK Normal
brusbmdm	Brother MFC USB Fax Only Modem					
mdm.sys	c:\windows\system32\drivers\brusbmdm.sys		Kernel Driver	No	Manual	Stopped OK Normal
brusbser	Brother MFC USB Serial WDM Driver					
ser.sys	c:\windows\system32\drivers\brusbser.sys		Kernel Driver	No	Manual	Stopped OK Normal
cdfs	CD/DVD File System Reader					
ys	c:\windows\system32\drivers\cdfs.sys		File System Driver	No	Disabled	Stopped OK Normal
cdrom	cdrom					
m.sys	c:\windows\system32\drivers\cdrom.sys		Kernel Driver	No	Manual	Stopped OK Normal
clfs	Common Log (CLFS)					
clfs.sys	c:\windows\system32\clfs.sys		Kernel Driver	Yes	Boot	Running OK Critical

cmbatt	Microsoft ACPI Control Method Battery Driver		
tt.sys	c:\windows\system32\drivers\cmba	Kernel Driver No	Manual
		Stopped OK	Normal
	No	No	
cmdide	cmdide		
	c:\windows\system32\drivers\cmdid	Kernel Driver No	Manual
e.sys		Stopped OK	Critical
	No	No	
cng	CNG		
	c:\windows\system32\drivers\cng.s	Kernel Driver Yes	Boot
ys		Running OK	Critical
	No	Yes	
compbatt	Compbatt		
	c:\windows\system32\drivers\comp	Kernel Driver No	Manual
batt.sys		Stopped OK	Critical
	No	No	
compositebus	Composite Bus		
Enumerator Driver			
	c:\windows\system32\drivers\comp	Kernel Driver Yes	Manual
ositebus.sys		Running OK	Normal
	No	Yes	
crdisk	Crdisk Filter Driver		
	c:\windows\system32\drivers\crdis	Kernel Driver No	Disabled
k.sys		Stopped OK	Normal
	No	No	
dfsc	DFS Namespace Client Driver		
	c:\windows\system32\drivers\dfsc.s	File System Driver	Yes
ys		System Running	OK
	No	No	Yes
discache	System Attribute Cache		
	c:\windows\system32\drivers\discac	Kernel Driver Yes	System
he.sys		Running OK	Normal
	No	Yes	
disk	Disk Driver		
	c:\windows\system32\drivers\disk.s	Kernel Driver Yes	Boot
ys		Running OK	Normal
	No	Yes	
dxgkrnl	LDDM Graphics Subsystem		
	c:\windows\system32\drivers\dxgkr	Kernel Driver No	Manual
nl.sys		Stopped OK	Ignore
	No	No	
ebdrv	Broadcom NetXtreme II 10 GigE		
VBD			
	c:\windows\system32\drivers\evbda	Kernel Driver Yes	Manual
.sys		Running OK	Normal
	No	Yes	
elxstor	elxstor		
	c:\windows\system32\drivers\elxsto	Kernel Driver No	Manual
r.sys		Stopped OK	Normal
	No	No	
errdev	Microsoft Hardware Error Device		
Driver			
	c:\windows\system32\drivers\errde	Kernel Driver No	Manual
v.sys		Stopped OK	Normal
	No	No	

exfat	exFAT File System Driver		
	c:\windows\system32\drivers\exfat.	File System Driver	No
sys		Manual Stopped	OK
	Normal	No	No
fastfat	FAT12/16/32 File System Driver		
	c:\windows\system32\drivers\fastfa	File System Driver	No
t.sys		Manual Stopped	OK
	Normal	No	No
fdc	Floppy Disk Controller Driver		
	c:\windows\system32\drivers\fdc.sy	Kernel Driver No	Manual
s		Stopped OK	Normal
	No	No	
fileinfo	File Information FS MiniFilter		
	c:\windows\system32\drivers\fileinf	File System Driver	No
o.sys		Manual Stopped	OK
	Normal	No	No
filetrace	Filetrace		
	c:\windows\system32\drivers\filetra	File System Driver	No
ce.sys		Manual Stopped	OK
	Normal	No	No
flpydisk	Floppy Disk Driver		
	c:\windows\system32\drivers\flpydi	Kernel Driver No	Manual
sk.sys		Stopped OK	Normal
	No	No	
ftmgr	FitMgr		
	c:\windows\system32\drivers\ftmgr	File System Driver	Yes
.sys		Boot Running	OK
	Critical	No	Yes
fsdepends	File System Dependency Minifilter		
	c:\windows\system32\drivers\fsdep	File System Driver	No
ends.sys		Manual Stopped	OK
	Critical	No	No
gagp30kx	Microsoft Generic AGPv3.0 Filter for K8 Processor Platforms		
	c:\windows\system32\drivers\gagp	Kernel Driver No	Manual
30kx.sys		Stopped OK	Normal
	No	No	
hdaudbus	Microsoft UAA Bus Driver for High Definition Audio		
	c:\windows\system32\drivers\hdau	Kernel Driver No	Manual
dbus.sys		Stopped OK	Normal
	No	No	
hidbatt	HID UPS Battery Driver		
	c:\windows\system32\drivers\hidba	Kernel Driver No	Manual
tt.sys		Stopped OK	Normal
	No	No	
hidusb	Microsoft HID Class Driver		
	c:\windows\system32\drivers\hidus	Kernel Driver Yes	Manual
b.sys		Running OK	Ignore
	No	Yes	
hpsamd	HpSAMD		
	c:\windows\system32\drivers\hpsa	Kernel Driver Yes	Boot
md.sys		Running OK	Normal
	No	Yes	

http	HTTP		
	c:\windows\system32\drivers\http.s	Kernel Driver Yes	Manual
ys		Running OK	Normal
hwpolicy	Hardware Policy Driver		
	c:\windows\system32\drivers\hwpol	Kernel Driver Yes	Boot
icy.sys		Running OK	Normal
	No	Yes	
i8042prt	i8042 Keyboard and PS/2 Mouse		
Port Driver			
	c:\windows\system32\drivers\i8042	Kernel Driver No	Manual
prt.sys		Stopped OK	Normal
	No	No	
iastorv	iaStorV		
	c:\windows\system32\drivers\iastor	Kernel Driver No	Manual
v.sys		Stopped OK	Normal
	No	No	
iirsp	iirsp		
	c:\windows\system32\drivers\iirsp.s	Kernel Driver No	Manual
ys		Stopped OK	Normal
	No	No	
intelide	intelide		
	c:\windows\system32\drivers\inteli	Kernel Driver No	Manual
e.sys		Stopped OK	Critical
	No	No	
intelppm	Intel Processor Driver		
	c:\windows\system32\drivers\intelp	Kernel Driver Yes	Manual
pm.sys		Running OK	Normal
	No	Yes	
ioatdma	Intel(R) QuickData Technology		
Device			
	c:\windows\system32\drivers\qd26	Kernel Driver No	Manual
0x64.sys		Stopped OK	Normal
	No	No	
ipfilterdriver	IP Traffic Filter Driver		
	c:\windows\system32\drivers\ipflt	Kernel Driver No	Manual
v.sys		Stopped OK	Normal
	No	No	
ipmidrv	IPMIDRV		
	c:\windows\system32\drivers\ipmid	Kernel Driver Yes	Manual
rv.sys		Running OK	Normal
	No	Yes	
ipnat	IP Network Address Translator		
	c:\windows\system32\drivers\ipnat.	Kernel Driver No	Manual
sys		Stopped OK	Normal
	No	No	
isapnp	isapnp		
	c:\windows\system32\drivers\isapn	Kernel Driver No	Manual
p.sys		Stopped OK	Critical
	No	No	
iscsiprt	iScsiPort Driver		
	c:\windows\system32\drivers\msisc	Kernel Driver No	Manual
si.sys		Stopped OK	Normal
	No	No	
kbdclass	Keyboard Class Driver		
	c:\windows\system32\drivers\kbdcl	Kernel Driver Yes	Manual
ass.sys		Running OK	Normal
	No	Yes	

kbdhid	Keyboard HID Driver
d.sys	Kernel Driver Yes Manual Running OK Ignore No Yes
ksecdd	KSecDD
d.sys	c:\windows\system32\drivers\ksecd Kernel Driver Yes Boot Running OK Critical No Yes
ksecpkg	KSecPkg
kg.sys	c:\windows\system32\drivers\ksecp Kernel Driver Yes Boot Running OK Critical No Yes
ksthunk	Kernel Streaming Thunks
nk.sys	c:\windows\system32\drivers\ksth Kernel Driver No Manual Stopped OK Normal No No
l2nd	Broadcom NetXtreme II BXND
60a.sys	c:\windows\system32\drivers\bxnd Kernel Driver Yes Manual Running OK Normal No Yes
ltdio	Link-Layer Topology Discovery Mapper I/O Driver
sys	c:\windows\system32\drivers\ltdio. Kernel Driver Yes Auto Running OK Normal No Yes
lsi_fc	LSI_FC
sys	c:\windows\system32\drivers\lsi_fc. Kernel Driver No Manual Stopped OK Normal No No
lsi_sas	LSI_SAS
s.sys	c:\windows\system32\drivers\lsi_sa Kernel Driver No Manual Stopped OK Normal No No
lsi_sas2	LSI_SAS2
s2.sys	c:\windows\system32\drivers\lsi_sa Kernel Driver No Manual Stopped OK Normal No No
lsi_scsi	LSI_SCSI
si.sys	c:\windows\system32\drivers\lsi_sc Kernel Driver No Manual Stopped OK Normal No No
luafv	UAC File Virtualization
sys	c:\windows\system32\drivers\luafv. File System Driver Yes Auto Running OK Normal No Yes
megasas	megasas
sas.sys	c:\windows\system32\drivers\mega Kernel Driver No Manual Stopped OK Normal No No
megasr	MegaSR
sr.sys	c:\windows\system32\drivers\mega Kernel Driver No Manual Stopped OK Normal No No
modem	Modem
m.sys	c:\windows\system32\drivers\mode Kernel Driver No Manual Stopped OK Ignore No No

monitor	Microsoft Monitor Class Function Driver Service
or.sys	c:\windows\system32\drivers\monit Kernel Driver Yes Manual Running OK Normal
mouclass	Mouse Class Driver
ass.sys	c:\windows\system32\drivers\moucl Kernel Driver Yes Manual Running OK Normal No Yes
mouhid	Mouse HID Driver
id.sys	c:\windows\system32\drivers\mouh Kernel Driver Yes Manual Running OK Ignore No Yes
mountmgr	Mount Point Manager
tmgr.sys	c:\windows\system32\drivers\moun Kernel Driver Yes Boot Running OK Critical No Yes
mpio	mpio
sys	c:\windows\system32\drivers\mpio. Kernel Driver No Manual Stopped OK Normal No No
mpsdrv	Windows Firewall Authorization Driver
rv.sys	c:\windows\system32\drivers\mpsd Kernel Driver Yes Manual Running OK Normal No Yes
mrxsm	SMB MiniRedirector Wrapper and Engine
mb.sys	c:\windows\system32\drivers\mrxs File System Driver Yes Manual Running OK Normal No Yes
mrxsm10	SMB 1.x MiniRedirector
mb10.sys	c:\windows\system32\drivers\mrxs File System Driver Yes Manual Running OK Normal No Yes
mrxsm20	SMB 2.0 MiniRedirector
mb20.sys	c:\windows\system32\drivers\mrxs File System Driver Yes Manual Running OK Normal No Yes
msahci	msahci
ci.sys	c:\windows\system32\drivers\msah Kernel Driver No Manual Stopped OK Critical No No
msdsm	msdsm
m.sys	c:\windows\system32\drivers\msds Kernel Driver No Manual Stopped OK Normal No No
msfs	Msfs
sys	c:\windows\system32\drivers\msfs. File System Driver Yes System Running OK Normal No Yes
mshidkmdf	Pass-through HID to KMDF Filter Driver
kmdf.sys	c:\windows\system32\drivers\mshid Kernel Driver No Manual Stopped OK Ignore No No

msisadrv	msisadrv
drv.sys	c:\windows\system32\drivers\msisa Kernel Driver Yes Boot Running OK Critical
msrpc	MsRPC
.sys	c:\windows\system32\drivers\msrpc Kernel Driver No Manual Stopped OK Normal No No
mssmbios	Microsoft System Management BIOS Driver
bios.sys	c:\windows\system32\drivers\mssm Kernel Driver Yes System Running OK Normal No Yes
mtconfig	Microsoft Input Configuration Driver
nfig.sys	c:\windows\system32\drivers\mtco Kernel Driver No Manual Stopped OK Normal No No
mup	Mup
sys	c:\windows\system32\drivers\mup. File System Driver Yes Boot Running OK Normal No Yes
ndis	NDIS System Driver
ys	c:\windows\system32\drivers\ndis.s Kernel Driver Yes Boot Running OK Critical No Yes
ndiscap	NDIS Capture LightWeight Filter
ap.sys	c:\windows\system32\drivers\ndisc Kernel Driver No Manual Stopped OK Normal No No
ndistapi	Remote Access NDIS TAPI Driver
pi.sys	c:\windows\system32\drivers\ndista Kernel Driver Yes Manual Running OK Normal No Yes
ndisui	NDIS Usermode I/O Protocol
o.sys	c:\windows\system32\drivers\ndisui Kernel Driver No Manual Stopped OK Normal No No
ndiswan	Remote Access NDIS WAN Driver
an.sys	c:\windows\system32\drivers\ndisw Kernel Driver Yes Manual Running OK Normal No Yes
ndproxy	NDIS Proxy
xy.sys	c:\windows\system32\drivers\ndpro Kernel Driver Yes Manual Running OK Normal No Yes
netbios	NetBIOS Interface
os.sys	c:\windows\system32\drivers\netbi File System Driver Yes System Running OK Normal No Yes
netbt	NetBT
sys	c:\windows\system32\drivers\netbt. Kernel Driver Yes System Running OK Normal No Yes
nfrd960	nfrd960
60.sys	c:\windows\system32\drivers\nfrd9 Kernel Driver No Manual Stopped OK Normal No No

npfs	Npfs		
	c:\windows\system32\drivers\npfs.s		
ys	File System Driver	Yes	
	System	Running	OK
	Normal	No	Yes
nsiproxy	NSI proxy service driver.		
	c:\windows\system32\drivers\nsipr		
oxy.sys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
ntfs	Ntfs		
	c:\windows\system32\drivers\ntfs.s		
ys	File System Driver	Yes	
	Manual	Running	OK
	Normal	No	Yes
null	Null		
	c:\windows\system32\drivers\null.s		
ys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
nvraid	nvraid		
	c:\windows\system32\drivers\nvrai		
d.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
nvstor	nvstor		
	c:\windows\system32\drivers\nvsto		
r.sys	Kernel Driver	No	Manual
	Stopped	OK	Critical
	No	No	
nv_agp	NVIDIA nForce AGP Bus Filter		
	c:\windows\system32\drivers\nv_ag		
p.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
ohci1394	1394 OHCI Compliant Host		
Controller (Legacy)			
	c:\windows\system32\drivers\ohci1		
394.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
parport	Parallel port driver		
	c:\windows\system32\drivers\parpo		
rt.sys	Kernel Driver	No	Manual
	Stopped	OK	Ignore
	No	No	
partmgr	Partition Manager		
	c:\windows\system32\drivers\partm		
gr.sys	Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
pci	PCI Bus Driver		
	c:\windows\system32\drivers\pci.sy		
s	Kernel Driver	Yes	Boot
	Running	OK	Critical
	No	Yes	
pciide	pciide		
	c:\windows\system32\drivers\pciide		
.sys	Kernel Driver	No	Manual
	Stopped	OK	Critical
	No	No	
pcmcia	pcmcia		
	c:\windows\system32\drivers\pcmcia		
a.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
pcw	Performance Counters for Windows		
Driver			
	c:\windows\system32\drivers\pcw.s		
ys	Kernel Driver	Yes	Boot
	Running	OK	Normal
	No	Yes	

peauth	PEAUTH		
	c:\windows\system32\drivers\peaut		
h.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
pptpminiport	WAN Miniport (PPTP)		
	c:\windows\system32\drivers\raspp		
tp.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
processor	Processor Driver		
	c:\windows\system32\drivers\proce		
ssr.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
psched	QoS Packet Scheduler		
	c:\windows\system32\drivers\pacer		
.sys	Kernel Driver	Yes	System
	Running	OK	Normal
	No	Yes	
ql2300	ql2300		
	c:\windows\system32\drivers\ql230		
0.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
ql40xx	ql40xx		
	c:\windows\system32\drivers\ql40x		
x.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
rasacd	Remote Access Auto Connection		
Driver			
	c:\windows\system32\drivers\rasac		
d.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
rasagilevpn	WAN Miniport (IKEv2)		
	c:\windows\system32\drivers\agilev		
pn.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasl2tp	WAN Miniport (L2TP)		
	c:\windows\system32\drivers\rasl2t		
p.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasppoe	Remote Access PPPOE Driver		
	c:\windows\system32\drivers\raspp		
poe.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rasstp	WAN Miniport (SSTP)		
	c:\windows\system32\drivers\rasst		
p.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rdbss	Redirected Buffering Sub System		
	c:\windows\system32\drivers\rdbss.		
sys	File System Driver	Yes	
	System	Running	OK
	Normal	No	Yes
rdpbus	Remote Desktop Device Redirector		
Bus Driver			
	c:\windows\system32\drivers\rdpbu		
s.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
rdpcdd	RDPCDD		
	c:\windows\system32\drivers\rdpcd		
d.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	

rdpdr	Terminal Server Device Redirector		
Driver			
	c:\windows\system32\drivers\rdpdr.		
sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
rdpencdd	RDP Encoder/Decoder Driver		
	c:\windows\system32\drivers\rdpen		
cdd.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
rdpprefmp	Reflector Display Driver used to		
gain access to graphics data			
	c:\windows\system32\drivers\rdpref		
mp.sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
rdpwd	RDP Winstation Driver		
	c:\windows\system32\drivers\rdpw		
d.sys	Kernel Driver	Yes	Manual
	Running	OK	Ignore
	No	Yes	
rspndr	Link-Layer Topology Discovery		
Responder			
	c:\windows\system32\drivers\rspnd		
r.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
	No	Yes	
s3cap	s3cap		
	c:\windows\system32\drivers\vms3		
cap.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
sacdrv	sacdrv		
	c:\windows\system32\drivers\sacdr		
v.sys	Kernel Driver	Yes	Boot
	Running	OK	Ignore
	No	Yes	
sbp2port	sbp2port		
	c:\windows\system32\drivers\sbp2p		
ort.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
scfilter	Smart card PnP Class Filter Driver		
	c:\windows\system32\drivers\scfite		
r.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
secdrv	Security Driver		
	c:\windows\system32\drivers\secdr		
v.sys	Kernel Driver	Yes	Auto
	Running	OK	Normal
	No	Yes	
serenum	Serenum Filter Driver		
	c:\windows\system32\drivers\seren		
um.sys	Kernel Driver	Yes	Manual
	Running	OK	Normal
	No	Yes	
serial	Serial port driver		
	c:\windows\system32\drivers\serial.		
sys	Kernel Driver	Yes	System
	Running	OK	Ignore
	No	Yes	
sermouse	Serial Mouse Driver		
	c:\windows\system32\drivers\serm		
ouse.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	
sfddisk	SFF Storage Class Driver		
	c:\windows\system32\drivers\sfddis		
k.sys	Kernel Driver	No	Manual
	Stopped	OK	Normal
	No	No	

sffp_mmc	SFF Storage Protocol Driver for MMC		
mmc.sys	c:\windows\system32\drivers\sffp_	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
sffp_sd	SFF Storage Protocol Driver for SDBus		
d.sys	c:\windows\system32\drivers\sffp_s	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
sfloppy	High-Capacity Floppy Disk Drive		
y.sys	c:\windows\system32\drivers\sflopp	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
sisraid2	SiSRaid2		
d2.sys	c:\windows\system32\drivers\sisrai	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
sisraid4	SiSRaid4		
d4.sys	c:\windows\system32\drivers\sisrai	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
smb	Message-oriented TCP/IP and TCP/IPV6 Protocol (SMB session)		
ys	c:\windows\system32\drivers\smb.s	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
spldr	Security Processor Loader Driver		
sys	c:\windows\system32\drivers\spldr.	Kernel Driver Yes	Boot
		Running OK	Critical
		No Yes	
srv	Server SMB 1.xxx Driver		
s	c:\windows\system32\drivers\srv.sy	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
srv2	Server SMB 2.xxx Driver		
ys	c:\windows\system32\drivers\srv2.s	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
srvnet	srvnet		
t.sys	c:\windows\system32\drivers\srve	File System Driver	Yes
		Manual Running	OK
		Normal No	Yes
stexstor	stexstor		
or.sys	c:\windows\system32\drivers\stext	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
storflt	Disk Virtual Machine Bus Acceleration Filter Driver		
rfl.sys	c:\windows\system32\drivers\vmsto	Kernel Driver Yes	Boot
		Running OK	Normal
		No Yes	
storvsc	storvsc		
c.sys	c:\windows\system32\drivers\storvs	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	

storvsp	storvsp		
p.sys	c:\windows\system32\drivers\storvs	Kernel Driver No	Manual
		Stopped OK	Normal
swenum	Software Bus Enumerator		
um.sys	c:\windows\system32\drivers\swen	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
tcpip	TCP/IP Protocol Driver		
sys	c:\windows\system32\drivers\tcpip.	Kernel Driver Yes	Boot
		Running OK	Normal
		No Yes	
tcpip6	Microsoft IPv6 Protocol Driver		
sys	c:\windows\system32\drivers\tcpip.	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
tcpipreg	TCP/IP Registry Compatibility		
eg.sys	c:\windows\system32\drivers\tcpipr	Kernel Driver Yes	Auto
		Running OK	Normal
		No Yes	
tdpipe	TDPIPE		
e.sys	c:\windows\system32\drivers\tdpip	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
tdtcp	TDTCP		
sys	c:\windows\system32\drivers\tdtcp.	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
tdx	NetIO Legacy TDI Support Driver		
s	c:\windows\system32\drivers\tdx.sy	Kernel Driver Yes	System
		Running OK	Normal
		No Yes	
termdd	Terminal Device Driver		
d.sys	c:\windows\system32\drivers\termd	Kernel Driver Yes	System
		Running OK	Normal
		No Yes	
tssecsrv	Remote Desktop Services Security Filter Driver		
rv.sys	c:\windows\system32\drivers\tssecs	Kernel Driver Yes	Manual
		Running OK	Ignore
		No Yes	
tunnel	Microsoft Tunnel Miniport Adapter Driver		
l.sys	c:\windows\system32\drivers\tunne	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
uagp35	Microsoft AGPv3.5 Filter		
35.sys	c:\windows\system32\drivers\uagp	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
udfs	udfs		
ys	c:\windows\system32\drivers\udfs.s	File System Driver	No
		Disabled Stopped	OK
		Normal No	No
uliagpkx	Uli AGP Bus Filter		
kx.sys	c:\windows\system32\drivers\uliagp	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	

umbus	UMBus Enumerator Driver		
s.sys	c:\windows\system32\drivers\umbu	Kernel Driver Yes	Manual
		Running OK	Normal
umpass	Microsoft UMPass Driver		
ss.sys	c:\windows\system32\drivers\umpa	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
usbccgp	Microsoft USB Generic Parent Driver		
gp.sys	c:\windows\system32\drivers\usbcc	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
usbhci	Microsoft USB 2.0 Enhanced Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usbh	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
usbhub	Microsoft USB Standard Hub Driver		
b.sys	c:\windows\system32\drivers\usbhu	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
usbohci	Microsoft USB Open Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usboh	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
usbprint	Microsoft USB PRINTER Class		
int.sys	c:\windows\system32\drivers\usbpr	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
usbstor	USB Mass Storage Driver		
or.sys	c:\windows\system32\drivers\usbst	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
usbuhci	Microsoft USB Universal Host Controller Miniport Driver		
ci.sys	c:\windows\system32\drivers\usbuh	Kernel Driver Yes	Manual
		Running OK	Normal
		No Yes	
vdrrvroot	Microsoft Virtual Drive Enumerator Driver		
oot.sys	c:\windows\system32\drivers\vdrrv	Kernel Driver Yes	Boot
		Running OK	Critical
		No Yes	
vga	vga		
p.sys	c:\windows\system32\drivers\vgapn	Kernel Driver Yes	Manual
		Running OK	Ignore
		No Yes	
vgasave	VgaSave		
ys	c:\windows\system32\drivers\vgas	Kernel Driver Yes	System
		Running OK	Ignore
		No Yes	
vhdmp	vhdmp		
p.sys	c:\windows\system32\drivers\vhdmp	Kernel Driver No	Manual
		Stopped OK	Normal
		No No	
viaide	viaide		
.sys	c:\windows\system32\drivers\viaide	Kernel Driver No	Manual
		Stopped OK	Critical
		No No	

```

vid Vid
c:\windows\system32\drivers\vid.sys
s Kernel Driver No Manual
Stopped OK Normal
No No
vmbus Virtual Machine Bus
c:\windows\system32\drivers\vmbru
s.sys Kernel Driver No Manual
Stopped OK Normal
No No
vmbushid VMBusHID
c:\windows\system32\drivers\vmbru
shid.sys Kernel Driver No Manual
Stopped OK Ignore
No No
volmgr Volume Manager Driver
c:\windows\system32\drivers\volmgr
r.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
volmgrx Dynamic Volume Manager
c:\windows\system32\drivers\volmgr
rx.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
volsnap Storage volumes
c:\windows\system32\drivers\volsnap
ap.sys Kernel Driver Yes Boot
Running OK Critical
No Yes
vsmraid vsmraid
c:\windows\system32\drivers\vsma
id.sys Kernel Driver No Manual
Stopped OK Normal
No No
wacompen Wacom Serial Pen HID Driver
c:\windows\system32\drivers\waco
mpen.sys Kernel Driver No Manual
Stopped OK Normal
No No
wanarp Remote Access IP ARP Driver
c:\windows\system32\drivers\wana
rp.sys Kernel Driver No Manual
Stopped OK Normal
No No
wanarpv6 Remote Access IPv6 ARP Driver
c:\windows\system32\drivers\wana
rp.sys Kernel Driver Yes System
Running OK Normal
No Yes
wd Wd
c:\windows\system32\drivers\wd.sys
s Kernel Driver No Manual
Stopped OK Normal
No No
wdf01000 Kernel Mode Driver Frameworks
service
c:\windows\system32\drivers\wdf0
1000.sys Kernel Driver Yes Boot
Running OK Normal
No Yes
wfpwfp WFP Lightweight Filter
c:\windows\system32\drivers\wfpwfp
f.sys Kernel Driver Yes System
Running OK Normal
No Yes
wimmount WIMMount
c:\windows\system32\drivers\wimm
ount.sys File System Driver No
Manual Stopped OK
Normal No No

```

```

wmiacpi Microsoft Windows Management
Interface for ACPI
c:\windows\system32\drivers\wmiacpi.sys
Kernel Driver No Manual
Stopped OK Normal
ws2ifsl Winsock IFS Driver
c:\windows\system32\drivers\ws2ifsl.sys
Kernel Driver No Disabled
Stopped OK Normal
No No
wudfppf User Mode Driver Frameworks
Platform Driver
c:\windows\system32\drivers\wudfppf.sys
Kernel Driver No Manual
Stopped OK Normal
No No

[Environment Variables]

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe
FP_NO_HOST_CHECK NO
OS Windows_NT <SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%\system32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\;C:\Program Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files (x86)\Microsoft SQL Server\100\DTS\Binn\;c:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JSE;.JSE;.WSF;.WSH;.MSC <SYSTEM>
PROCESSOR_ARCHITECTURE AMD64
TEMP %SystemRoot%\TEMP
TMP %SystemRoot%\TEMP
USERNAME SYSTEM <SYSTEM>
windir %SystemRoot%
PSModulePath %SystemRoot%\system32\WindowsPowerShell\v1.0\Modules\ <SYSTEM>
NUMBER_OF_PROCESSORS 8
PROCESSOR_LEVEL 6
PROCESSOR_IDENTIFIER Intel64 Family 6 Model 26 Stepping 5, GenuineIntel
PROCESSOR_REVISION 1a05
VS90COMNTOOLS c:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\Tools\ <SYSTEM>
TEMP %USERPROFILE%\AppData\Local\Temp
emp NT AUTHORITY\SYSTEM
TMP %USERPROFILE%\AppData\Local\Temp
emp NT AUTHORITY\SYSTEM

```

```

TEMP %USERPROFILE%\AppData\Local\Temp
emp NT AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\AppData\Local\Temp
emp NT AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\AppData\Local\Temp
emp SERVICE NT AUTHORITY\NETWORK
TMP %USERPROFILE%\AppData\Local\Temp
emp SERVICE NT AUTHORITY\NETWORK
TEMP %USERPROFILE%\AppData\Local\Temp
emp C7KBAY2\Administrator
TMP %USERPROFILE%\AppData\Local\Temp
emp C7KBAY2\Administrator
TEMP %USERPROFILE%\AppData\Local\Temp
emp IIS APPPOOL\Classic .NET AppPool
emp IIS APPPOOL\Classic .NET AppPool

[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

[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
Available Not Available Not Available Not Available
Available Not Available 8
system Not Available Not Available
5/11/2011 11:35 AM Not Available
Available Not Available Not Available
smss.exe Not Available 272 11
200 1380
5/11/2011 11:35 AM Not Available
Available Not Available Not Available
csrss.exe c:\windows\system32\csrss.exe
372 13 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 7.50 KB
(7,680 bytes) 7/13/2009 4:19 PM
wininit.exe c:\windows\system32\wininit.exe
424 13 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 126.00
KB (129,024 bytes) 7/13/2009 4:52 PM

```



```

csrss.exe c:\windows\system32\csrss.exe
432 13 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
services.exe c:\windows\system32\services.exe
480 9 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 321.00
KB (328,704 bytes) 7/13/2009 4:19 PM

lsass.exe c:\windows\system32\lsass.exe
488 9 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 30.50 KB
(31,232 bytes) 7/13/2009 4:20 PM

lsm.exe c:\windows\system32\lsm.exe
496 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 325.50
KB (333,312 bytes) 7/13/2009 5:17 PM

winlogon.exe c:\windows\system32\winlogon.exe
528 13 200
1380 5/11/2011 11:35 AM
6.1.7600.16447 380.50
KB (389,632 bytes) 9/15/2010 10:46 AM

svchost.exe c:\windows\system32\svchost.exe
628 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
708 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

logonui.exe c:\windows\system32\logonui.exe
796 13 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 27.00 KB
(27,648 bytes) 7/13/2009 4:52 PM

svchost.exe c:\windows\system32\svchost.exe
804 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
852 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
900 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

svchost.exe c:\windows\system32\svchost.exe
944 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

```

```

svchost.exe c:\windows\system32\svchost.exe
988 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM
svchost.exe c:\windows\system32\svchost.exe
364 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

spoolsv.exe c:\windows\system32\spoolsv.exe
1036 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16661 545.50
KB (558,592 bytes) 9/15/2010 11:40 AM

svchost.exe c:\windows\system32\svchost.exe
1072 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

inetinfo.exe c:\windows\system32\inetinfo.exe
1096 8 200
1380 5/11/2011 11:35 AM
7.5.7600.16385 15.00 KB
(15,360 bytes) 7/13/2009 5:27 PM

sqlservr.exe c:\program files (x86)\microsoft sql
server\mssql.1\bin\sqlservr.exe 1184
8 200 1380
5/11/2011 11:35 AM
2005.90.3042.0 27.83
MB (29,178,224 bytes) 2/10/2007 5:29 AM

svchost.exe c:\windows\system32\svchost.exe
1256 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

rsys.exe c:\program files
(x86)\benchcraft\rsys.exe 1400 8
200 1380
5/11/2011 11:35 AM Not
Available 9.50 KB (9,728 bytes)
4/19/2006 6:44 PM
sqlwriter.exe c:\program files\microsoft sql
server\90\shared\sqlwriter.exe 1428
8 200 1380
5/11/2011 11:35 AM
2005.90.3042.0 152.36
KB (156,016 bytes) 2/10/2007 10:03 AM

svchost.exe c:\windows\system32\svchost.exe
1456 8 200
1380 5/11/2011 11:35 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

dagent.exe c:\program
files\altiris\client\dagent.exe 1564
8 200 1380
5/11/2011 11:35 AM
6.9.430.0 1.85 MB (1,937,232
bytes) 8/11/2009 5:48 PM
svchost.exe c:\windows\system32\svchost.exe
1244 8 200
1380 5/11/2011 11:36 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM

```

```

svchost.exe c:\windows\system32\svchost.exe
2060 8 200
1380 5/11/2011 11:36 AM
6.1.7600.16385 26.50 KB
(27,136 bytes) 7/13/2009 4:31 PM
msdtc.exe c:\windows\system32\msdtc.exe
2592 8 200
1380 5/11/2011 11:38 AM
2001.12.8530.16385 138.50
KB (141,824 bytes) 7/13/2009 4:59 PM

dllhost.exe c:\windows\system32\dllhost.exe
2156 8 200
1380 5/11/2011 11:53 AM
6.1.7600.16385 9.50 KB
(9,728 bytes)7/13/2009 4:59 PM
csrss.exe c:\windows\system32\csrss.exe
2424 13 200
1380 5/13/2011 12:12 PM
6.1.7600.16385 7.50 KB
(7,680 bytes)7/13/2009 4:19 PM
winlogon.exe c:\windows\system32\winlogon.exe
2360 13 200
1380 5/13/2011 12:12 PM
6.1.7600.16447 380.50
KB (389,632 bytes) 9/15/2010 10:46 AM

taskhost.exe c:\windows\system32\taskhost.exe
2980 8 200
1380 5/13/2011 12:12 PM
6.1.7600.16385 67.50 KB
(69,120 bytes) 7/13/2009 4:31 PM

rdpclip.exe c:\windows\system32\rdpclip.exe
2668 8 200
1380 5/13/2011 12:12 PM
6.1.7600.16385 204.50
KB (209,408 bytes) 7/13/2009 5:17 PM

dwm.exe c:\windows\system32\dwm.exe
2428 8 200
1380 5/13/2011 12:12 PM
6.1.7600.16385 117.50
KB (120,320 bytes) 7/13/2009 4:37 PM

explorer.exe c:\windows\explorer.exe 2524
8 200 1380
5/13/2011 12:12 PM
6.1.7600.16450 2.74 MB
(2,870,272 bytes) 9/15/2010 10:46 AM

dagentui.exe c:\program
files\altiris\client\dagentui.exe 596
8 200 1380
5/13/2011 12:12 PM
6.9.430.0 827.83 KB (847,696
bytes) 8/11/2009 5:48 PM
msinfo32.exe c:\windows\system32\msinfo32.exe
3060 8 200
1380 5/16/2011 2:43 PM
6.1.7600.16385 370.00
KB (378,880 bytes) 7/13/2009 4:31 PM

wmiprvse.exe c:\windows\system32\wbem\wmipr
vse.exe 920 8 200
1380 5/16/2011 2:43 PM
6.1.7600.16385 360.00
KB (368,640 bytes) 7/13/2009 4:47 PM

```

wmiprvse.exe
 c:\windows\system32\wbem\wmiprvse.exe
 772 8 200
 1380 5/16/2011 2:43 PM
 6.1.7600.16385 360.00
 KB (368,640 bytes) 7/13/2009 4:47 PM

[Loaded Modules]

Name	Version	Size	File Date
csrss	6.1.7600.16385	7.50 KB	7/13/2009 4:19 PM
Microsoft Corporation c:\windows\system32\csrss.exe			
ntdll	6.1.7600.16559	1.66 MB	9/15/2010 10:45 AM
(1,736,608 bytes) Microsoft Corporation c:\windows\system32\ntdll.dll			
csrsrv	6.1.7600.16385	42.50 KB	7/13/2009 4:19 PM
(43,520 bytes) Microsoft Corporation c:\windows\system32\csrsrv.dll			
basesrv	6.1.7600.16385	51.50 KB	7/13/2009 4:18 PM
(52,736 bytes) Microsoft Corporation c:\windows\system32\basesrv.dll			
winsrv	6.1.7600.16385	209.00 KB	7/13/2009 4:38 PM
(214,016 bytes) Microsoft Corporation c:\windows\system32\winsrv.dll			
user32	6.1.7600.16385	985.00 KB	7/13/2009 4:38 PM
(1,008,640 bytes) Microsoft Corporation c:\windows\system32\user32.dll			
gdi32	6.1.7600.16385	395.00 KB	7/13/2009 4:39 PM
(404,480 bytes) Microsoft Corporation c:\windows\system32\gdi32.dll			
kernel32	6.1.7600.16385	1.11 MB	7/13/2009 4:28 PM
(1,162,240 bytes) Microsoft Corporation c:\windows\system32\kernel32.dll			
kernelbase	6.1.7600.16385	411.50 KB	7/13/2009 4:20 PM
(421,376 bytes) Microsoft Corporation c:\windows\system32\kernelbase.dll			
lpk	6.1.7600.16385	41.00 KB	7/13/2009 4:38 PM
(41,984 bytes) Microsoft Corporation c:\windows\system32\lpk.dll			
usp10	6.1.7600.16385	782.50 KB	7/13/2009 4:38 PM
(801,280 bytes) Microsoft Corporation c:\windows\system32\usp10.dll			
msvcrt	7.0.7600.16385	620.00 KB	7/13/2009 4:19 PM
(634,880 bytes) Microsoft Corporation c:\windows\system32\msvcrt.dll			

sxssrv	6.1.7600.16385	31.00 KB	7/13/2009 4:26 PM
(31,744 bytes) Microsoft Corporation c:\windows\system32\sxssrv.dll			
sxs	6.1.7600.16385	569.50 KB	7/13/2009 4:27 PM
(583,168 bytes) Microsoft Corporation c:\windows\system32\sxs.dll			
rpcrt4	6.1.7600.16385	1.17 MB	7/13/2009 4:23 PM
(1,221,632 bytes) Microsoft Corporation c:\windows\system32\rpcrt4.dll			
cryptbase	6.1.7600.16385	43.00 KB	7/13/2009 4:20 PM
(44,032 bytes) Microsoft Corporation c:\windows\system32\cryptbase.dll			
wininit	6.1.7600.16385	126.00 KB	7/13/2009 4:52 PM
(129,024 bytes) Microsoft Corporation c:\windows\system32\wininit.exe			
sechost	6.1.7600.16385	111.00 KB	7/13/2009 4:20 PM
(113,664 bytes) Microsoft Corporation c:\windows\system32\sechost.dll			
profapi	6.1.7600.16385	43.00 KB	7/13/2009 4:20 PM
(44,032 bytes) Microsoft Corporation c:\windows\system32\profapi.dll			
imm32	6.1.7600.16385	163.50 KB	7/13/2009 4:38 PM
(167,424 bytes) Microsoft Corporation c:\windows\system32\imm32.dll			
msctf	6.1.7600.16385	1.02 MB	7/13/2009 4:40 PM
(1,067,008 bytes) Microsoft Corporation c:\windows\system32\msctf.dll			
rpcrtremote	6.1.7600.16385	63.50 KB	7/13/2009 4:59 PM
(65,024 bytes) Microsoft Corporation c:\windows\system32\rpcrtremote.dll			
apphelp	6.1.7600.16385	330.50 KB	7/13/2009 4:21 PM
(338,432 bytes) Microsoft Corporation c:\windows\system32\apphelp.dll			
ws2_32	6.1.7600.16385	289.50 KB	7/13/2009 4:21 PM
(296,448 bytes) Microsoft Corporation c:\windows\system32\ws2_32.dll			
nsi	6.1.7600.16385	13.50 KB	7/13/2009 4:21 PM
(13,824 bytes) Microsoft Corporation c:\windows\system32\nsi.dll			
msocket	6.1.7600.16385	312.50 KB	7/13/2009 4:21 PM
(320,000 bytes) Microsoft Corporation c:\windows\system32\msocket.dll			
wshtcpip	6.1.7600.16385	13.00 KB	7/13/2009 4:21 PM
(13,312 bytes) Microsoft Corporation c:\windows\system32\wshtcpip.dll			

wship6	6.1.7600.16385	13.50 KB	7/13/2009 4:21 PM
(13,824 bytes) Microsoft Corporation c:\windows\system32\wship6.dll			
secur32	6.1.7600.16385	27.50 KB	7/13/2009 4:50 PM
(28,160 bytes) Microsoft Corporation c:\windows\system32\secur32.dll			
sspicli	6.1.7600.16385	133.00 KB	7/13/2009 4:20 PM
(136,192 bytes) Microsoft Corporation c:\windows\system32\sspicli.dll			
credssp	6.1.7600.16385	20.00 KB	7/13/2009 4:50 PM
(20,480 bytes) Microsoft Corporation c:\windows\system32\credssp.dll			
advapi32	6.1.7600.16385	856.50 KB	7/13/2009 5:41 PM
(877,056 bytes) Microsoft Corporation c:\windows\system32\advapi32.dll			
services	6.1.7600.16385	321.00 KB	7/13/2009 4:19 PM
(328,704 bytes) Microsoft Corporation c:\windows\system32\services.exe			
sceext	6.1.7600.16385	87.00 KB	7/13/2009 4:31 PM
(89,088 bytes) Microsoft Corporation c:\windows\system32\sceext.dll			
scsvr	6.1.7600.16385	396.50 KB	7/13/2009 4:49 PM
(406,016 bytes) Microsoft Corporation c:\windows\system32\scsvr.dll			
svcli	6.1.7600.16385	124.50 KB	7/13/2009 4:53 PM
(127,488 bytes) Microsoft Corporation c:\windows\system32\svcli.dll			
authz	6.1.7600.16385	173.50 KB	7/13/2009 4:50 PM
(177,664 bytes) Microsoft Corporation c:\windows\system32\authz.dll			
ubpm	6.1.7600.16385	209.00 KB	7/13/2009 4:31 PM
(214,016 bytes) Microsoft Corporation c:\windows\system32\ubpm.dll			
wtsapi32	6.1.7600.16385	53.00 KB	7/13/2009 5:17 PM
(54,272 bytes) Microsoft Corporation c:\windows\system32\wtsapi32.dll			
winsta	6.1.7600.16385	228.00 KB	7/13/2009 5:17 PM
(233,472 bytes) Microsoft Corporation c:\windows\system32\winsta.dll			
lsass	6.1.7600.16385	30.50 KB	7/13/2009 4:20 PM
(31,232 bytes) Microsoft Corporation c:\windows\system32\lsass.exe			
sspsrv	6.1.7600.16385	28.00 KB	7/13/2009 4:20 PM
(28,672 bytes) Microsoft Corporation c:\windows\system32\sspsrv.dll			

lsasrv 6.1.7600.16484 1.38 MB
(1,446,912 bytes) 9/15/2010 10:46 AM
Microsoft Corporation
c:\windows\system32\lsasrv.dll

samsrv 6.1.7600.16385 740.00
KB (757,760 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\samsrv.dll

cryptdll 6.1.7600.16385 64.50 KB
(66,048 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cryptdll.dll

msasn1 6.1.7600.16415 45.50 KB
(46,592 bytes) 9/15/2010 11:40 AM
Microsoft Corporation
c:\windows\system32\msasn1.dll

wevtapi 6.1.7600.16385 418.00
KB (428,032 bytes) 7/13/2009 4:46 PM
Microsoft Corporation
c:\windows\system32\wevtapi.dll

cngaudit 6.1.7600.16385 18.50 KB
(18,944 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cngaudit.dll

ncrypt 6.1.7600.16385 300.00
KB (307,200 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\ncrypt.dll

bcrypt 6.1.7600.16385 121.00
KB (123,904 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\bcrypt.dll

msprivs 6.1.7600.16385 2.00 KB
(2,048 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\msprivs.dll

netjoin 6.1.7600.16385 184.50
KB (188,928 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netjoin.dll

negoexts 6.1.7600.16385 114.50
KB (117,248 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\negoexts.dll

kerberos 6.1.7600.16385 697.50
KB (714,240 bytes) 7/13/2009 4:51 PM
Microsoft Corporation
c:\windows\system32\kerberos.dll

cryptsp 6.1.7600.16385 78.00 KB
(79,872 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\cryptsp.dll

msv1_0 6.1.7600.16420 304.50
KB (311,808 bytes) 9/15/2010 11:21 AM
Microsoft Corporation
c:\windows\system32\msv1_0.dll

netlogon 6.1.7600.16385 676.50
KB (692,736 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netlogon.dll

dnsapi 6.1.7600.16385 348.00
KB (356,352 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dnsapi.dll

logoncli 6.1.7600.16385 182.00
KB (186,368 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\logoncli.dll

schannel 6.1.7600.16612 333.00
KB (340,992 bytes) 9/15/2010 10:46 AM
Microsoft Corporation
c:\windows\system32\schannel.dll

crypt32 6.1.7600.16385 1.39 MB
(1,454,592 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\crypt32.dll

wdigest 6.1.7600.16385 205.50
KB (210,432 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\wdigest.dll

rsaenh 6.1.7600.16385 274.66
KB (281,256 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\rsaenh.dll

tspkg 6.1.7600.16385 84.00 KB
(86,016 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\tspkg.dll

pku2u 6.1.7600.16385 235.00
KB (240,640 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\pku2u.dll

bcryptprimitives 6.1.7600.16385
291.32 KB (298,312 bytes)
7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\bcryptprimitives.dll

es.dll 6.1.7600.16385 55.50 KB
(56,832 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\eflslaext.dll

scecli 6.1.7600.16385 227.00
KB (232,448 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\scecli.dll

rassfm 6.1.7600.16385 28.50 KB
(29,184 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\rassfm.dll

mpr 6.1.7600.16385 79.00 KB
(80,896 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\mpr.dll

netutils 6.1.7600.16385 28.00 KB
(28,672 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netutils.dll

userenv 6.1.7600.16385 104.50
KB (107,008 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\userenv.dll

samcli 6.1.7600.16385 65.50 KB
(67,072 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\samcli.dll

samlib 6.1.7600.16385 104.50
KB (107,008 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\samlib.dll

dssenh 6.1.7600.16385 186.41
KB (190,880 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\dssenh.dll

gpapi 6.1.7600.16385 94.50 KB
(96,768 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\gpapi.dll

cryptnet 6.1.7600.16385 135.50
KB (138,752 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\cryptnet.dll

wldap32 6.1.7600.16385 304.50
KB (311,808 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\wldap32.dll

certpoleng 6.1.7600.16385 70.00 KB
(71,680 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\certpoleng.dll

iphlpapi 6.1.7600.16385 142.50
KB (145,920 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\iphlpapi.dll

winnsi 6.1.7600.16385 25.50 KB
(26,112 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\winnsi.dll

ism 6.1.7600.16385 325.50
KB (333,312 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\ism.exe

sysntfy 6.1.7600.16385 22.50 KB
(23,040 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\sysntfy.dll

wmsgapi 6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 4:52 PM
Microsoft Corporation
c:\windows\system32\wmsgapi.dll

pcwum 6.1.7600.16385 36.00 KB
(36,864 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\pcwum.dll

ole32 6.1.7600.16385 1.99 MB
(2,084,352 bytes) 7/13/2009 5:02 PM
Microsoft Corporation
c:\windows\system32\ole32.dll

ntmarta 6.1.7600.16385 158.50
KB (162,304 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\ntmarta.dll

clbcattq	2001.12.8530.16385	593.50 KB (607,744 bytes)	7/13/2009 5:00 PM	Microsoft Corporation	c:\windows\system32\clbcattq.dll
oleaut32	6.1.7600.16567	841.00 KB (861,184 bytes)	9/15/2010 10:45 AM	Microsoft Corporation	c:\windows\system32\oleaut32.dll
lsmpoxy	6.1.7600.16385	47.50 KB (48,640 bytes)	7/13/2009 5:17 PM	Microsoft Corporation	c:\windows\system32\lsmpoxy.dll
winlogon	6.1.7600.16447	380.50 KB (389,632 bytes)	9/15/2010 10:46 AM	Microsoft Corporation	c:\windows\system32\winlogon.exe
uxinit	6.1.7600.16385	24.50 KB (25,088 bytes)	7/13/2009 4:54 PM	Microsoft Corporation	c:\windows\system32\uxinit.dll
svchost	6.1.7600.16385	26.50 KB (27,136 bytes)	7/13/2009 4:31 PM	Microsoft Corporation	c:\windows\system32\svchost.exe
umpnpgm	6.1.7600.16385	395.00 KB (404,480 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\umpnpgm.dll
ls					
spinf	6.1.7600.16385	103.00 KB (105,472 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\spinf.dll
devrtl	6.1.7600.16385	57.00 KB (58,368 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\devrtl.dll
umpo	6.1.7600.16385	160.00 KB (163,840 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\umpo.dll
setupapi	6.1.7600.16385	1.81 MB (1,899,520 bytes)	7/13/2009 4:27 PM	Microsoft Corporation	c:\windows\system32\setupapi.dll
cfgmgr32	6.1.7600.16385	202.50 KB (207,360 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\cfgmgr32.dll
devobj	6.1.7600.16385	91.00 KB (93,184 bytes)	7/13/2009 4:26 PM	Microsoft Corporation	c:\windows\system32\devobj.dll
rpcss	6.1.7600.16385	497.50 KB (509,440 bytes)	7/13/2009 5:00 PM	Microsoft Corporation	c:\windows\system32\rpcss.dll
wmidcpv	6.1.7600.16385	187.00 KB (191,488 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\wmidc
prv.dll					

fastprox	6.1.7600.16385	888.00 KB (909,312 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\fastpr
wbemcomn	6.1.7600.16385	517.50 KB (529,920 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbemcomn.d
ll					
ntdsapi	6.1.7600.16385	148.50 KB (152,064 bytes)	7/13/2009 4:54 PM	Microsoft Corporation	c:\windows\system32\ntdsapi.dll
wbemprox	6.1.7600.16385	42.50 KB (43,520 bytes)	7/13/2009 4:46 PM	Microsoft Corporation	c:\windows\system32\wbem\wbem
prox.dll					
wbemsvc	6.1.7600.16385	63.00 KB (64,512 bytes)	7/13/2009 4:46 PM	Microsoft Corporation	c:\windows\system32\wbem\wbem
svc.dll					
wmiutils	6.1.7600.16385	134.00 KB (137,216 bytes)	7/13/2009 4:47 PM	Microsoft Corporation	c:\windows\system32\wbem\wmiuti
ls.dll					
wintrust	6.1.7600.16493	215.50 KB (220,672 bytes)	9/15/2010 10:46 AM	Microsoft Corporation	c:\windows\system32\wintrust.dll
rpcepmap	6.1.7600.16385	65.50 KB (67,072 bytes)	7/13/2009 4:21 PM	Microsoft Corporation	c:\windows\system32\rpcepmap.dll
firewallapi	6.1.7600.16385	730.50 KB (748,032 bytes)	7/13/2009 5:08 PM	Microsoft Corporation	c:\windows\system32\firewallapi.dll
version	6.1.7600.16385	28.50 KB (29,184 bytes)	7/13/2009 4:57 PM	Microsoft Corporation	c:\windows\system32\version.dll
fwpuclnt	6.1.7600.16385	316.50 KB (324,096 bytes)	7/13/2009 5:09 PM	Microsoft Corporation	c:\windows\system32\fwpuclnt.dll
logonui	6.1.7600.16385	27.00 KB (27,648 bytes)	7/13/2009 4:52 PM	Microsoft Corporation	c:\windows\system32\logonui.exe
authui	6.1.7600.16385	1.84 MB (1,926,144 bytes)	7/13/2009 4:58 PM	Microsoft Corporation	c:\windows\system32\authui.dll
cryptui	6.1.7600.16385	1.02 MB (1,065,984 bytes)	7/13/2009 4:49 PM	Microsoft Corporation	c:\windows\system32\cryptui.dll

comctl32	6.10.7600.16385	1.94 MB (2,030,080 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\winsxs\amd64_microsoft
t.windows.common-					
controls_6595b64144ccf1df_6.0.7600.16385_x-ww					
shlwapi	6.1.7600.16385	439.00 KB (449,536 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\shlwapi.dll
uxtheme	6.1.7600.16385	324.50 KB (332,288 bytes)	7/13/2009 4:55 PM	Microsoft Corporation	c:\windows\system32\uxtheme.dll
gdipplus	6.1.7600.16385	2.06 MB (2,165,248 bytes)	7/13/2009 4:40 PM	Microsoft Corporation	c:\windows\winsxs\amd64_microsoft
t.windows.gdipplus_6595b64144ccf1df_1.1.7600.16385_none_2b4f45e87195fcc4\gdipplus.dll					
dui70	6.1.7600.16385	954.00 KB (976,896 bytes)	7/13/2009 4:41 PM	Microsoft Corporation	c:\windows\system32\dui70.dll
duser	6.1.7600.16385	254.50 KB (260,608 bytes)	7/13/2009 4:39 PM	Microsoft Corporation	c:\windows\system32\duser.dll
sndvol32	6.1.7600.16385	220.00 KB (225,280 bytes)	7/13/2009 5:19 PM	Microsoft Corporation	c:\windows\system32\sndvol32.dll
hid	6.1.7600.16385	29.50 KB (30,208 bytes)	7/13/2009 5:06 PM	Microsoft Corporation	c:\windows\system32\hid.dll
mmdevapi	6.1.7600.16385	277.50 KB (284,160 bytes)	7/13/2009 5:18 PM	Microsoft Corporation	c:\windows\system32\mmdevapi.dll
propsys	7.0.7600.16385	1.16 MB (1,212,416 bytes)	7/13/2009 4:56 PM	Microsoft Corporation	c:\windows\system32\propsys.dll
dwmapi	6.1.7600.16385	80.50 KB (82,432 bytes)	7/13/2009 4:37 PM	Microsoft Corporation	c:\windows\system32\dwmapi.dll
xmllite	1.3.1000.0	195.00 KB (199,680 bytes)	7/13/2009 5:41 PM	Microsoft Corporation	c:\windows\system32\xmllite.dll
windowscodecs	6.1.7600.16385	1.13 MB (1,189,888 bytes)	7/13/2009 4:42 PM	Microsoft Corporation	c:\windows\system32\windowscode
cs.dll					
winbrand	6.1.7600.16385	16.00 KB (16,384 bytes)	7/13/2009 4:30 PM	Microsoft Corporation	c:\windows\system32\winbrand.dll

vaultcredprovider 6.1.7600.16385
78.50 KB (80,384 bytes)
7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\vaultcredprov
ider.dll
smartcardcredentialprovider
6.1.7600.16385 185.50
KB (189,952 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\smartcardcre
dentialprovider.dll
certcredprovider 6.1.7600.16385
126.00 KB (129,024 bytes)
7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\certcredprovi
der.dll
raslap 6.1.7600.16385 396.00
KB (405,504 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\raslap.dll
rasapi32 6.1.7600.16385 375.50
KB (384,512 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\rasapi32.dll
rasman 6.1.7600.16385 98.00 KB
(100,352 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\rasman.dll
rtutils 6.1.7600.16617 51.00 KB
(52,224 bytes) 9/15/2010 10:45 AM
Microsoft Corporation
c:\windows\system32\rtutils.dll
shacct 6.1.7600.16385 132.00
KB (135,168 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\system32\shacct.dll
wevtvc 6.1.7600.16385 1.57 MB
(1,646,080 bytes) 7/13/2009 4:49 PM
Microsoft Corporation
c:\windows\system32\wevtvc.dll
lmhsvc 6.1.7600.16385 23.00 KB
(23,552 bytes) 7/13/2009 5:09 PM
Microsoft Corporation
c:\windows\system32\lmhsvc.dll
nrpsrv 6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 5:09 PM
Microsoft Corporation
c:\windows\system32\nrpsrv.dll
dhcpcore 6.1.7600.16385 307.00
KB (314,368 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dhcpcore.dll
dhcpcore6 6.1.7600.16385 219.00
KB (224,256 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dhcpcore6.dll
dhcpcsvc6 6.1.7600.16385 53.00 KB
(54,272 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dhcpcsvc6.dll

dhcpcsvc 6.1.7600.16385 85.00 KB
(87,040 bytes) 7/13/2009 4:21 PM
Microsoft Corporation
c:\windows\system32\dhcpcsvc.dll
nlaapi 6.1.7600.16385 68.50 KB
(70,144 bytes) 7/13/2009 5:09 PM
Microsoft Corporation
c:\windows\system32\nlaapi.dll
rasadhlp 6.1.7600.16385 16.00 KB
(16,384 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\rasadhlp.dll
winmr 6.1.7600.16385 28.00 KB
(28,672 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\winmr.dll
napinsp 6.1.7600.16385 66.50 KB
(68,096 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\napinsp.dll
gpsvc 6.1.7600.16385 758.00
KB (776,192 bytes) 7/13/2009 4:54 PM
Microsoft Corporation
c:\windows\system32\gpsvc.dll
profsvc 6.1.7600.16385 203.50
KB (208,384 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\profsvc.dll
atl 3.5.2284.0 88.50 KB (90,624
bytes) 7/13/2009 5:34 PM
Microsoft Corporation
c:\windows\system32\atl.dll
dsrole 6.1.7600.16385 32.00 KB
(32,768 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\dsrole.dll
slc 6.1.7600.16385 30.00 KB
(30,720 bytes) 7/13/2009 4:51 PM
Microsoft Corporation
c:\windows\system32\slc.dll
sens 6.1.7600.16385 63.00 KB
(64,512 bytes) 7/13/2009 4:34 PM
Microsoft Corporation
c:\windows\system32\sens.dll
schedsvc 6.1.7600.16385 1.05 MB
(1,104,384 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\schedsvc.dll
shell32 6.1.7600.16644 13.51
MB (14,162,944 bytes) 9/15/2010 10:46 AM
Microsoft Corporation
c:\windows\system32\shell32.dll
netapi32 6.1.7600.16385 71.00 KB
(72,704 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\netapi32.dll
wksccli 6.1.7600.16385 70.00 KB
(71,680 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\wksccli.dll

ktmw32 6.1.7600.16385 22.50 KB
(23,040 bytes) 7/13/2009 4:19 PM
Microsoft Corporation
c:\windows\system32\ktmw32.dll
taskcomp 6.1.7600.16385 462.50
KB (473,600 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\taskcomp.dll
ikeext 6.1.7600.16385 826.00
KB (845,824 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\ikeext.dll
sacsvr 6.1.7600.16385 14.50 KB
(14,848 bytes) 7/13/2009 4:34 PM
Microsoft Corporation
c:\windows\system32\sacsvr.dll
wmisvc 6.1.7600.16385 237.00
KB (242,688 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\wmisv
c.dll
vssapi 6.1.7600.16385 1.66 MB
(1,745,408 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\vssapi.dll
vsstrace 6.1.7600.16385 75.00 KB
(76,800 bytes) 7/13/2009 4:36 PM
Microsoft Corporation
c:\windows\system32\vsstrace.dll
wbemcore 6.1.7600.16385 1.16 MB
(1,220,096 bytes) 7/13/2009 4:48 PM
Microsoft Corporation
c:\windows\system32\wbem\wbem
core.dll
esscli 6.1.7600.16385 430.00
KB (440,320 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\esscli.
dll
repdrvfs 6.1.7600.16385 441.00
KB (451,584 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\repdrv
fs.dll
wmiprvsd 6.1.7600.16385 732.50
KB (750,080 bytes) 7/13/2009 4:48 PM
Microsoft Corporation
c:\windows\system32\wbem\wmi
prvsd.dll
ncobjapi 6.1.7600.16385 67.50 KB
(69,120 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\ncobjapi.dll
wbemess 6.1.7600.16385 494.00
KB (505,856 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\wbem
ess.dll
iphlpvc 6.1.7600.16385 552.50
KB (565,760 bytes) 7/13/2009 5:09 PM
Microsoft Corporation
c:\windows\system32\iphlpvc.dll
sqmapi 6.1.7600.16385 229.50
KB (235,008 bytes) 7/13/2009 4:40 PM
Microsoft Corporation
c:\windows\system32\sqmapi.dll

wdscore 6.1.7600.16385 265.00 KB (271,360 bytes) 7/13/2009 4:28 PM
 Microsoft Corporation
 c:\windows\system32\wdscore.dll

srsvsvc 6.1.7600.16385 230.00 KB (235,520 bytes) 7/13/2009 4:53 PM
 Microsoft Corporation
 c:\windows\system32\srsvsvc.dll

browser 6.1.7600.16385 133.00 KB (136,192 bytes) 7/13/2009 4:53 PM
 Microsoft Corporation
 c:\windows\system32\browser.dll

sscore 6.1.7600.16385 13.00 KB (13,312 bytes) 7/13/2009 4:53 PM
 Microsoft Corporation
 c:\windows\system32\sscore.dll

clusapi 6.1.7600.16385 307.00 KB (314,368 bytes) 7/13/2009 4:34 PM
 Microsoft Corporation
 c:\windows\system32\clusapi.dll

resutils 6.1.7600.16385 84.00 KB (86,016 bytes) 7/13/2009 4:34 PM
 Microsoft Corporation
 c:\windows\system32\resutils.dll

netprofm 6.1.7600.16385 449.00 KB (459,776 bytes) 7/13/2009 5:12 PM
 Microsoft Corporation
 c:\windows\system32\netprofm.dll

nci 6.1.7600.16385 87.50 KB (89,600 bytes) 7/13/2009 5:09 PM
 Microsoft Corporation
 c:\windows\system32\nci.dll

npmproxy 6.1.7600.16385 31.00 KB (31,744 bytes) 7/13/2009 5:12 PM
 Microsoft Corporation
 c:\windows\system32\npmproxy.dll

certprop 6.1.7600.16385 78.50 KB (80,384 bytes) 7/13/2009 4:50 PM
 Microsoft Corporation
 c:\windows\system32\certprop.dll

winscard 6.1.7600.16385 212.50 KB (217,600 bytes) 7/13/2009 4:50 PM
 Microsoft Corporation
 c:\windows\system32\winscard.dll

sessenv 6.1.7600.16385 102.50 KB (104,960 bytes) 7/13/2009 5:17 PM
 Microsoft Corporation
 c:\windows\system32\sessenv.dll

ncprov 6.1.7600.16385 76.50 KB (78,336 bytes) 7/13/2009 4:47 PM
 Microsoft Corporation
 c:\windows\system32\wbem\ncprov.dll

qmgr 7.5.7600.16385 828.50 KB (848,384 bytes) 7/13/2009 4:46 PM
 Microsoft Corporation
 c:\windows\system32\qmgr.dll

bitsperf 7.5.7600.16385 23.50 KB (24,064 bytes) 7/13/2009 4:46 PM
 Microsoft Corporation
 c:\windows\system32\bitsperf.dll

bitsigd 7.5.7600.16385 55.50 KB (56,832 bytes) 7/13/2009 4:46 PM
 Microsoft Corporation
 c:\windows\system32\bitsigd.dll

winhttp 6.1.7600.16385 428.50 KB (438,784 bytes) 7/13/2009 5:11 PM
 Microsoft Corporation
 c:\windows\system32\winhttp.dll

webio 6.1.7600.16385 385.50 KB (394,752 bytes) 7/13/2009 5:11 PM
 Microsoft Corporation
 c:\windows\system32\webio.dll

wuaueng 7.3.7600.16385 2.31 MB (2,418,176 bytes) 7/13/2009 5:36 PM
 Microsoft Corporation
 c:\windows\system32\wuaueng.dll

esent 6.1.7600.16385 2.45 MB (2,565,120 bytes) 7/13/2009 4:50 PM
 Microsoft Corporation
 c:\windows\system32\esent.dll

winspool 6.1.7600.16385 431.50 KB (441,856 bytes) 7/13/2009 5:39 PM
 Microsoft Corporation
 c:\windows\system32\winspool.drv

cabinet 6.1.7600.16385 92.00 KB (94,208 bytes) 7/13/2009 4:21 PM
 Microsoft Corporation
 c:\windows\system32\cabinet.dll

mspatcha 6.1.7600.16385 45.50 KB (46,592 bytes) 7/13/2009 4:21 PM
 Microsoft Corporation
 c:\windows\system32\mspatcha.dll

psapi 6.1.7600.16385 9.00 KB (9,216 bytes) 7/13/2009 4:26 PM
 Microsoft Corporation
 c:\windows\system32\psapi.dll

wer 6.1.7600.16385 473.00 KB (484,352 bytes) 7/13/2009 4:41 PM
 Microsoft Corporation
 c:\windows\system32\wer.dll

msi 5.0.7600.16385 3.06 MB (3,211,776 bytes) 7/13/2009 4:51 PM
 Microsoft Corporation
 c:\windows\system32\msi.dll

advpack 8.0.7600.16385 156.50 KB (160,256 bytes) 7/13/2009 4:58 PM
 Microsoft Corporation
 c:\windows\system32\advpack.dll

msxml3 8.110.7600.16605 1.79 MB (1,877,504 bytes) 9/15/2010 10:45 AM
 Microsoft Corporation
 c:\windows\system32\msxml3.dll

shsvcs 6.1.7600.16385 361.00 KB (369,664 bytes) 7/13/2009 4:55 PM
 Microsoft Corporation
 c:\windows\system32\shsvcs.dll

es 2001.12.8530.16385 393.50 KB (402,944 bytes) 7/13/2009 5:00 PM
 Microsoft Corporation
 c:\windows\system32\es.dll

actxprxy 6.1.7600.16385 936.50 KB (958,976 bytes) 7/13/2009 5:41 PM
 Microsoft Corporation
 c:\windows\system32\actxprxy.dll

hnetcfg 6.1.7600.16385 414.50 KB (424,448 bytes) 7/13/2009 5:08 PM
 Microsoft Corporation
 c:\windows\system32\hnetcfg.dll

aelupsvc 6.1.7600.16385 70.50 KB (72,192 bytes) 7/13/2009 4:21 PM
 Microsoft Corporation
 c:\windows\system32\aelupsvc.dll

nsisvc 6.1.7600.16385 25.00 KB (25,600 bytes) 7/13/2009 4:21 PM
 Microsoft Corporation
 c:\windows\system32\nsisvc.dll

w32time 6.1.7600.16385 373.00 KB (381,952 bytes) 7/13/2009 4:49 PM
 Microsoft Corporation
 c:\windows\system32\w32time.dll

vmactimeprovider 6.1.7600.16385 50.50 KB (51,712 bytes) 7/13/2009 4:43 PM
 Microsoft Corporation
 c:\windows\system32\vmactimeprov

uler.dll 6.1.7600.16385 38.00 KB (38,912 bytes) 7/13/2009 4:37 PM
 Microsoft Corporation
 c:\windows\system32\uxsms.dll

uxsms 6.1.7600.16385 38.00 KB (38,912 bytes) 7/13/2009 4:37 PM
 Microsoft Corporation
 c:\windows\system32\uxsms.dll

trkwks 6.1.7600.16385 117.00 KB (119,808 bytes) 7/13/2009 4:59 PM
 Microsoft Corporation
 c:\windows\system32\trkwks.dll

umrdp 6.1.7600.16385 190.50 KB (195,072 bytes) 7/13/2009 5:18 PM
 Microsoft Corporation
 c:\windows\system32\umrdp.dll

umb 6.1.7600.16385 58.50 KB (59,904 bytes) 7/13/2009 4:35 PM
 Microsoft Corporation
 c:\windows\system32\umb.dll

printui 6.1.7600.16385 1.00 MB (1,049,600 bytes) 7/13/2009 5:40 PM
 Microsoft Corporation
 c:\windows\system32\printui.dll

puiapi 6.1.7600.16385 190.00 KB (194,560 bytes) 7/13/2009 5:40 PM
 Microsoft Corporation
 c:\windows\system32\puiapi.dll

netman 6.1.7600.16385 352.00 KB (360,448 bytes) 7/13/2009 5:08 PM
 Microsoft Corporation
 c:\windows\system32\netman.dll

netshell 6.1.7600.16385 2.53 MB (2,651,136 bytes) 7/13/2009 5:09 PM
 Microsoft Corporation
 c:\windows\system32\netshell.dll

rasdlg 6.1.7600.16385 840.50 KB (860,672 bytes) 7/13/2009 5:10 PM
 Microsoft Corporation
 c:\windows\system32\rasdlg.dll

mprapi 6.1.7600.16385 215.50 KB (220,672 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\mprapi.dll

netcfgx 6.1.7600.16385 505.00 KB (517,120 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\netcfgx.dll

dnssrslvr 6.1.7600.16385 178.00 KB (182,272 bytes) 7/13/2009 4:21 PM Microsoft Corporation c:\windows\system32\dnssrslvr.dll

dnsexth 6.1.7600.16385 8.00 KB (8,192 bytes) 7/13/2009 5:12 PM Microsoft Corporation c:\windows\system32\dnsexth.dll

wkssvc 6.1.7600.16385 116.00 KB (118,784 bytes) 7/13/2009 4:53 PM Microsoft Corporation c:\windows\system32\wkssvc.dll

cryptsvc 6.1.7600.16385 171.00 KB (175,104 bytes) 7/13/2009 4:49 PM Microsoft Corporation c:\windows\system32\cryptsvc.dll

nlascv 6.1.7600.16385 295.00 KB (302,080 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\nlascv.dll

ncsi 6.1.7600.16385 204.50 KB (209,408 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\ncsi.dll

ssdpapi 6.1.7600.16385 50.00 KB (51,200 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\ssdpapi.dll

wsmcvc 6.1.7600.16385 1.93 MB (2,018,816 bytes) 7/13/2009 4:49 PM Microsoft Corporation c:\windows\system32\wsmcvc.dll

httpapi 6.1.7600.16385 44.00 KB (45,056 bytes) 7/13/2009 4:21 PM Microsoft Corporation c:\windows\system32\httpapi.dll

wevtfwd 6.1.7600.16385 114.00 KB (116,736 bytes) 7/13/2009 4:46 PM Microsoft Corporation c:\windows\system32\wevtfwd.dll

bfe 6.1.7600.16385 687.00 KB (703,488 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\bfe.dll

mpssvc 6.1.7600.16385 805.50 KB (824,832 bytes) 7/13/2009 5:09 PM Microsoft Corporation c:\windows\system32\mpssvc.dll

wfapigp 6.1.7600.16385 22.00 KB (22,528 bytes) 7/13/2009 5:08 PM Microsoft Corporation c:\windows\system32\wfapigp.dll

dps 6.1.7600.16385 159.00 KB (162,816 bytes) 7/13/2009 4:31 PM Microsoft Corporation c:\windows\system32\dps.dll

taskschd 6.1.7600.16385 1.11 MB (1,168,896 bytes) 7/13/2009 4:47 PM Microsoft Corporation c:\windows\system32\taskschd.dll

wdi 6.1.7600.16385 88.50 KB (90,624 bytes) 7/13/2009 4:31 PM Microsoft Corporation c:\windows\system32\wdi.dll

wdiasqmmodule 6.1.7600.16385 35.00 KB (35,840 bytes) 7/13/2009 4:40 PM Microsoft Corporation c:\windows\system32\wdiasqmmodule.dll

ule.dll 6.1.7600.16385 95.50 KB (97,792 bytes) 7/13/2009 4:32 PM Microsoft Corporation c:\windows\system32\ule.dll

spoolsv 6.1.7600.16661 545.50 KB (558,592 bytes) 9/15/2010 11:40 AM Microsoft Corporation c:\windows\system32\spoolsv.exe

powrprof 6.1.7600.16385 163.50 KB (167,424 bytes) 7/13/2009 4:27 PM Microsoft Corporation c:\windows\system32\powrprof.dll

localspl 6.1.7600.16385 932.50 KB (954,880 bytes) 7/13/2009 5:40 PM Microsoft Corporation c:\windows\system32\localspl.dll

spoolss 6.1.7600.16385 56.50 KB (57,856 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\spoolss.dll

printisolationproxy 6.1.7600.16385 47.00 KB (48,128 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\printisolationproxy.dll

tcpmon 6.1.7600.16385 190.50 KB (195,072 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\tcpmon.dll

snmpapi 6.1.7600.16385 27.00 KB (27,648 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\snmpapi.dll

wsnmp32 6.1.7600.16385 65.50 KB (67,072 bytes) 7/13/2009 5:10 PM Microsoft Corporation c:\windows\system32\wsnmp32.dll

msxml6 6.30.7600.16385 1.91 MB (1,999,360 bytes) 7/13/2009 5:43 PM Microsoft Corporation c:\windows\system32\msxml6.dll

usbmon 6.1.7600.16385 44.00 KB (45,056 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\usbmon.dll

wls0wndh 6.1.7600.16385 10.50 KB (10,752 bytes) 7/13/2009 4:52 PM Microsoft Corporation c:\windows\system32\wls0wndh.dll

wsdmon 6.1.7600.16385 219.50 KB (224,768 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\wsdmon.dll

wsdapi 6.1.7600.16385 571.50 KB (585,216 bytes) 7/13/2009 4:36 PM Microsoft Corporation c:\windows\system32\wsdapi.dll

webservicess 6.1.7600.16385 1.11 MB (1,159,168 bytes) 7/13/2009 5:01 PM Microsoft Corporation c:\windows\system32\webservicess.dll

fundisc 6.1.7600.16385 190.00 KB (194,560 bytes) 7/13/2009 4:35 PM Microsoft Corporation c:\windows\system32\fundisc.dll

fdpnp 6.1.7600.16385 50.00 KB (51,200 bytes) 7/13/2009 4:35 PM Microsoft Corporation c:\windows\system32\fdpnp.dll

winprint 6.1.7600.16385 38.50 KB (39,424 bytes) 7/13/2009 5:39 PM Microsoft Corporation c:\windows\system32\spool\prtprocs\x64\winprint.dll

win32spl 6.1.7600.16385 728.50 KB (745,984 bytes) 7/13/2009 5:40 PM Microsoft Corporation c:\windows\system32\win32spl.dll

cscapi 6.1.7600.16385 45.00 KB (46,080 bytes) 7/13/2009 4:24 PM Microsoft Corporation c:\windows\system32\cscapi.dll

apphostsv 7.5.7600.16385 64.00 KB (65,536 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetrv\apphostsv.dll

iisutil 7.5.7600.16385 274.50 KB (281,088 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetrv\iisutil.dll

nativerd 7.5.7600.16385 458.50 KB (469,504 bytes) 7/13/2009 5:27 PM Microsoft Corporation c:\windows\system32\inetrv\nativerd.dll

iisres 7.5.7600.16385 215.00 KB (220,160 bytes) 7/13/2009 5:26 PM Microsoft Corporation c:\windows\system32\inetrv\iisres.dll

mlang 6.1.7600.16385 221.50 KB (226,816 bytes) 7/13/2009 4:55 PM Microsoft Corporation c:\windows\system32\mlang.dll

inetinfo 7.5.7600.16385 15.00 KB
(15,360 bytes) 7/13/2009 5:27 PM
Microsoft Corporation
c:\windows\system32\inetrv\inetin

fo.exe
rpcref 7.5.7600.16385 9.00 KB
(9,216 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\rpcref

.dll
iisrtl 7.5.7600.16385 187.00
KB (191,488 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\iisrtl.dll

iisadmin 7.5.7600.16385 23.50 KB
(24,064 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\iisad

min.dll
coadmin 7.5.7600.16385 81.50 KB
(83,456 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\coad

min.dll
admwprox 7.5.7600.16385 54.00 KB
(55,296 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\admwprox.dll

iiscfg 7.5.7600.16385 1.01 MB
(1,057,792 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\iiscfg.

dll
abocomp 7.5.7600.16385 215.00
KB (220,160 bytes) 7/13/2009 5:27 PM
Microsoft Corporation
c:\windows\system32\inetrv\aboco

mp.dll
wsock32 6.1.7600.16385 18.00 KB
(18,432 bytes) 7/13/2009 5:10 PM
Microsoft Corporation
c:\windows\system32\wsock32.dll

metadata 7.5.7600.16385 277.00
KB (283,648 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\meta

data.dll
svcext 7.5.7600.16385 16.50 KB
(16,896 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\svcex

t.dll
wamreg 7.5.7600.16385 36.50 KB
(37,376 bytes) 7/13/2009 5:26 PM
Microsoft Corporation
c:\windows\system32\inetrv\wamr

eg.dll
sqlservr 2005.90.3042.0 27.83
MB (29,178,224 bytes) 2/10/2007 5:29 AM
Microsoft Corporation
c:\program files (x86)\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe

wow64 6.1.7600.16491 237.50
KB (243,200 bytes) 9/15/2010 10:45 AM
Microsoft Corporation
c:\windows\system32\wow64.dll

wow64win 6.1.7600.16385 353.50
KB (361,984 bytes) 7/13/2009 4:38 PM
Microsoft Corporation
c:\windows\system32\wow64win.dll

wow64cpu 6.1.7600.16385 13.00 KB
(13,312 bytes) 7/13/2009 4:26 PM
Microsoft Corporation
c:\windows\system32\wow64cpu.dll

regsvc 6.1.7600.16385 155.50
KB (159,232 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\regsvc.dll

rsys Not Available 9.50 KB (9,728 bytes)
4/19/2006 6:44 PM Not
Available c:\program files
(x86)\benchmark\rsys.exe

sqlwriter 2005.90.3042.0 152.36
KB (156,016 bytes) 2/10/2007 10:03 AM
Microsoft Corporation
c:\program files\microsoft sql
server\90\shared\sqlwriter.exe

msvcr80 8.0.50727.4927 783.81
KB (802,624 bytes) 7/13/2009 1:37 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft
t.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4927_n
one_88dce9872fb18caf\msvcr80.dll

sqlwvss 2005.90.3042.0 365.86
KB (374,640 bytes) 2/10/2007 10:03 AM
Microsoft Corporation
c:\program files\microsoft sql
server\90\shared\sqlwvss.dll

msvcp80 8.0.50727.4927 1.02 MB
(1,068,352 bytes) 7/13/2009 1:37 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft
t.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.4927_n
one_88dce9872fb18caf\msvcp80.dll

iisw3adm 7.5.7600.16385 440.50
KB (451,072 bytes) 7/13/2009 5:27 PM
Microsoft Corporation
c:\windows\system32\inetrv\iisw3

adm.dll
w3tp 7.5.7600.16385 19.50 KB
(19,968 bytes) 7/13/2009 5:27 PM
Microsoft Corporation
c:\windows\system32\inetrv\w3tp.

dll
dagent 6.9.430.0 1.85 MB (1,937,232
bytes) 8/11/2009 5:48 PM Altiris,
Inc. c:\program
files\altiris\client\dagent.exe

termsrv 6.1.7600.16385 690.00
KB (706,560 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\termsrv.dll

icaapi 6.1.7600.16385 22.00 KB
(22,528 bytes) 7/13/2009 5:16 PM
Microsoft Corporation
c:\windows\system32\icaapi.dll

regapi 6.1.7600.16385 92.50 KB
(94,720 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\regapi.dll

tlscsp 6.1.7600.16385 72.00 KB
(73,728 bytes) 7/13/2009 5:16 PM
Microsoft Corporation
c:\windows\system32\tlscsp.dll

rdpcorekmts 6.1.7600.16385 146.00
KB (149,504 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\rdpcorekmts.

dll

rdpwsx 6.1.7600.16385 74.50 KB
(76,288 bytes) 7/13/2009 5:17 PM
Microsoft Corporation
c:\windows\system32\rdpwsx.dll

ipsevc 6.1.7600.16385 488.50
KB (500,224 bytes) 7/13/2009 5:08 PM
Microsoft Corporation
c:\windows\system32\ipsevc.dll

fwremotesvr 6.1.7600.16385 74.00 KB
(75,776 bytes) 7/13/2009 5:08 PM
Microsoft Corporation
c:\windows\system32\fwremotesvr.

dll
msdtc 2001.12.8530.16385 138.50
KB (141,824 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtc.exe

msdtctm 2001.12.8530.16385 1.44 MB
(1,509,888 bytes) 7/13/2009 5:00 PM
Microsoft Corporation
c:\windows\system32\msdtctm.dll

msdtcprx 2001.12.8530.16385 728.00
KB (745,472 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtcprx.dll

mtxclu 2001.12.8530.16385 364.00
KB (372,736 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\mtxclu.dll

msdtclog 2001.12.8530.16385 122.00
KB (124,928 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtclog.dll

winmm 6.1.7600.16385 212.50
KB (217,600 bytes) 7/13/2009 5:18 PM
Microsoft Corporation
c:\windows\system32\winmm.dll

xolehlp 2001.12.8530.16385 58.00 KB
(59,392 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\xolehlp.dll

comres 2001.12.8530.16385 1.24 MB
(1,297,408 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\comres.dll

msdtcvsp1res 2001.12.8530.16385
21.00 KB (21,504 bytes)
7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\msdtcvsp1res

.dll
mtxoci 2001.12.8530.16385 153.00
KB (156,672 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\mtxoci.dll

dllhost 6.1.7600.16385 9.50 KB
(9,728 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\dllhost.exe

comsvcs 2001.12.8530.16385 1.66 MB
(1,735,680 bytes) 7/13/2009 5:01 PM
Microsoft Corporation
c:\windows\system32\comsvcs.dll

bflog 2001.12.8530.16385 116.50
 KB (119,296 bytes) 7/13/2009 4:59 PM
 Microsoft Corporation
 c:\windows\system32\bflog.dll

catsrv 2001.12.8530.16385 461.50
 KB (472,576 bytes) 7/13/2009 5:00 PM
 Microsoft Corporation
 c:\windows\system32\catsrv.dll

mfcsubs 2001.12.8530.16385 33.00 KB
 (33,792 bytes) 7/13/2009 4:59 PM
 Microsoft Corporation
 c:\windows\system32\mfcsubs.dll

catsrvps 2001.12.8530.16385 55.00 KB
 (56,320 bytes) 7/13/2009 4:59 PM
 Microsoft Corporation
 c:\windows\system32\catsrvps.dll

catsrvut 2001.12.8530.16385 513.50
 KB (525,824 bytes) 7/13/2009 5:00 PM
 Microsoft Corporation
 c:\windows\system32\catsrvut.dll

taskhost 6.1.7600.16385 67.50 KB
 (69,120 bytes) 7/13/2009 4:31 PM
 Microsoft Corporation
 c:\windows\system32\taskhost.exe

msctfmonitor 6.1.7600.16385 27.50 KB
 (28,160 bytes) 7/13/2009 4:39 PM
 Microsoft Corporation
 c:\windows\system32\msctfmonitor
 .dll

msutb 6.1.7600.16385 230.00
 KB (235,520 bytes) 7/13/2009 4:39 PM
 Microsoft Corporation
 c:\windows\system32\msutb.dll

dimsjob 6.1.7600.16385 39.50 KB
 (40,448 bytes) 7/13/2009 4:53 PM
 Microsoft Corporation
 c:\windows\system32\dimsjob.dll

rdpcap 6.1.7600.16385 204.50
 KB (209,408 bytes) 7/13/2009 5:17 PM
 Microsoft Corporation
 c:\windows\system32\rdpcap.exe

dwm 6.1.7600.16385 117.50
 KB (120,320 bytes) 7/13/2009 4:37 PM
 Microsoft Corporation
 c:\windows\system32\dwm.exe

dwmredir 6.1.7600.16385 125.50
 KB (128,512 bytes) 7/13/2009 4:37 PM
 Microsoft Corporation
 c:\windows\system32\dwmredir.dll

dwmcore 6.1.7600.16385 1.56 MB
 (1,634,304 bytes) 7/13/2009 4:39 PM
 Microsoft Corporation
 c:\windows\system32\dwmcore.dll

d3d10_1 6.1.7600.16385 192.50
 KB (197,120 bytes) 7/13/2009 4:41 PM
 Microsoft Corporation
 c:\windows\system32\d3d10_1.dll

d3d10_1core 6.1.7600.16385 311.50
 KB (318,976 bytes) 7/13/2009 4:41 PM
 Microsoft Corporation
 c:\windows\system32\d3d10_1core.
 dll

dxgi 6.1.7600.16385 643.00
 KB (658,432 bytes) 7/13/2009 4:41 PM
 Microsoft Corporation
 c:\windows\system32\dxgi.dll

explorer 6.1.7600.16450 2.74 MB
 (2,870,272 bytes) 9/15/2010 10:46 AM
 Microsoft Corporation
 c:\windows\explorer.exe

explorerframe 6.1.7600.16385
 1.78 MB (1,863,680 bytes)
 7/13/2009 4:57 PM
 Microsoft Corporation
 c:\windows\system32\explorerfram
 e.dll

ehstorshell 6.1.7600.16385 198.50
 KB (203,264 bytes) 7/13/2009 5:00 PM
 Microsoft Corporation
 c:\windows\system32\ehstorshell.dl
 l

ntshrui 6.1.7600.16385 498.00
 KB (509,952 bytes) 7/13/2009 4:57 PM
 Microsoft Corporation
 c:\windows\system32\ntshrui.dll

iconcodecservice 6.1.7600.16385
 14.00 KB (14,336 bytes)
 7/13/2009 4:37 PM
 Microsoft Corporation
 c:\windows\system32\iconcodecserv
 ice.dll

timedate 6.1.7600.16385 503.00
 KB (515,072 bytes) 7/13/2009 4:56 PM
 Microsoft Corporation
 c:\windows\system32\timedate.cpl

shdocvw 6.1.7600.16385 191.50
 KB (196,096 bytes) 7/13/2009 4:55 PM
 Microsoft Corporation
 c:\windows\system32\shdocvw.dll

linkinfo 6.1.7600.16385 29.00 KB
 (29,696 bytes) 7/13/2009 4:55 PM
 Microsoft Corporation
 c:\windows\system32\linkinfo.dll

msftedit 5.41.21.2509 781.00 KB (799,744
 bytes) 7/13/2009 4:39 PM
 Microsoft Corporation
 c:\windows\system32\msftedit.dll

msils31 3.10.349.0 217.00 KB (222,208
 bytes) 7/13/2009 4:39 PM
 Microsoft Corporation
 c:\windows\system32\msils31.dll

urlmon 8.0.7600.16625 1.43 MB
 (1,494,528 bytes) 9/15/2010 10:46 AM
 Microsoft Corporation
 c:\windows\system32\urlmon.dll

iertutil 8.0.7600.16666 2.33 MB
 (2,441,216 bytes) 9/15/2010 12:53 PM
 Microsoft Corporation
 c:\windows\system32\iertutil.dll

msiltcfg 5.0.7600.16385 19.50 KB
 (19,968 bytes) 7/13/2009 4:48 PM
 Microsoft Corporation
 c:\windows\system32\msiltcfg.dll

stobject 6.1.7600.16385 250.00
 KB (256,000 bytes) 7/13/2009 4:56 PM
 Microsoft Corporation
 c:\windows\system32\stobject.dll

batmeter 6.1.7600.16385 730.50
 KB (748,032 bytes) 7/13/2009 4:56 PM
 Microsoft Corporation
 c:\windows\system32\batmeter.dll

prnfdrr 6.1.7600.16385 407.00
 KB (416,768 bytes) 7/13/2009 5:40 PM
 Microsoft Corporation
 c:\windows\system32\prnfdrr.dll

dxp 6.1.7600.16385 449.00
 KB (459,776 bytes) 7/13/2009 5:21 PM
 Microsoft Corporation
 c:\windows\system32\dxp.dll

syncreg 2007.94.7600.16385 72.00 KB
 (73,728 bytes) 7/13/2009 5:22 PM
 Microsoft Corporation
 c:\windows\system32\syncreg.dll

pnidui 6.1.7600.16385 1.72 MB
 (1,807,872 bytes) 7/13/2009 5:08 PM
 Microsoft Corporation
 c:\windows\system32\pnidui.dll

qutil 6.1.7600.16385 105.00
 KB (107,520 bytes) 7/13/2009 5:07 PM
 Microsoft Corporation
 c:\windows\system32\qutil.dll

actioncenter 6.1.7600.16385 762.50
 KB (780,800 bytes) 7/13/2009 4:56 PM
 Microsoft Corporation
 c:\windows\system32\actioncenter.
 dll

imapi2 6.1.7600.16385 493.50
 KB (505,344 bytes) 7/13/2009 5:01 PM
 Microsoft Corporation
 c:\windows\system32\imapi2.dll

hgcp1 6.1.7600.16385 324.50
 KB (332,288 bytes) 7/13/2009 4:57 PM
 Microsoft Corporation
 c:\windows\system32\hgcp1.dll

qagent 6.1.7600.16385 259.00
 KB (265,216 bytes) 7/13/2009 5:07 PM
 Microsoft Corporation
 c:\windows\system32\qagent.dll

ieproxy 8.0.7600.16625 440.50
 KB (451,072 bytes) 9/15/2010 10:46 AM
 Microsoft Corporation
 c:\program files\internet
 explorer\ieproxy.dll

drprov 6.1.7600.16385 24.00 KB
 (24,576 bytes) 7/13/2009 5:17 PM
 Microsoft Corporation
 c:\windows\system32\drprov.dll

ntlanman 6.1.7600.16385 126.50
 KB (129,536 bytes) 7/13/2009 4:48 PM
 Microsoft Corporation
 c:\windows\system32\ntlanman.dll

thumbcache 6.1.7600.16385 110.50
 KB (113,152 bytes) 7/13/2009 4:55 PM
 Microsoft Corporation
 c:\windows\system32\thumbcache.
 dll

ehstorapi 6.1.7600.16385 141.50
 KB (144,896 bytes) 7/13/2009 5:00 PM
 Microsoft Corporation
 c:\windows\system32\ehstorapi.dll

```

ieframe 8.0.7600.16625 11.79
MB (12,364,800 bytes) 9/15/2010 10:46 AM
Microsoft Corporation
c:\windows\system32\ieframe.dll

oleacc 7.0.0.0 324.00 KB (331,776
bytes) 7/13/2009 4:39 PM
Microsoft Corporation
c:\windows\system32\oleacc.dll

networkexplorer 6.1.7600.16385
1.60 MB (1,672,704 bytes)
7/13/2009 5:08 PM
Microsoft Corporation
c:\windows\system32\networkexplor

rer.dll

structuredquery 7.0.7600.16385
472.50 KB (483,840 bytes)
7/13/2009 5:29 PM
Microsoft Corporation
c:\windows\system32\structuredqu

ery.dll

werconcp 6.1.7600.16385 1.22 MB
(1,280,512 bytes) 7/13/2009 4:41 PM
Microsoft Corporation
c:\windows\system32\werconcp.dll

framedynos 6.1.7600.16385 288.50
KB (295,424 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\framedynos.d

ll

werclpsupport 6.1.7600.16385
82.50 KB (84,480 bytes)
7/13/2009 4:40 PM
Microsoft Corporation
c:\windows\system32\werclpsuppor

t.dll

hcproviders 6.1.7600.16385 30.50 KB
(31,232 bytes) 7/13/2009 4:56 PM
Microsoft Corporation
c:\windows\system32\hcproviders.d

ll

searchfolder 6.1.7600.16385 845.00
KB (865,280 bytes) 7/13/2009 4:59 PM
Microsoft Corporation
c:\windows\system32\searchfolder.

dll

naturallanguage6 6.1.7600.16385
1.27 MB (1,326,592 bytes)
7/13/2009 5:35 PM
Microsoft Corporation
c:\windows\system32\naturallangua

ge6.dll

nlsdata0009 6.1.7600.16385 5.98 MB
(6,270,976 bytes) 7/13/2009 5:31 PM
Microsoft Corporation
c:\windows\system32\nlsdata0009.

dll

nlslexicons0009 6.1.7600.16385
2.51 MB (2,628,608 bytes)
7/13/2009 5:33 PM
Microsoft Corporation
c:\windows\system32\nlslexicons00

09.dll

wininet 8.0.7600.16625 1.14 MB
(1,192,960 bytes) 9/15/2010 10:46 AM
Microsoft Corporation
c:\windows\system32\wininet.dll

normaliz 6.1.7600.16385 2.50 KB
(2,560 bytes) 7/13/2009 4:26 PM
Microsoft Corporation
c:\windows\system32\normaliz.dll

```

```

dagentui 6.9.430.0 827.83 KB (847,696
bytes) 8/11/2009 5:48 PM Altiris,
Inc.
c:\program
files\altiris\agent\dagentui.exe
581.50
KB (595,456 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\system32\comdlg32.dll

comctl32 5.82.7600.16385 619.00
KB (633,856 bytes) 7/13/2009 4:55 PM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft

t.windows.common-
controls_6595b64144cc1df_5.82.7600.16385_no
ne_a44af8ec57f961cf\comctl32.dll

dagentui_EN 6.9.430.0 416.34 KB (426,328
bytes) 8/11/2009 5:48 PM Altiris,
Inc.
c:\program
files\altiris\agent\dagentui_en.dll

msinfo32 6.1.7600.16385 370.00
KB (378,880 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\msinfo32.exe

mfc42u 6.6.8063.0 1.29 MB (1,357,312
bytes) 7/13/2009 5:35 PM
Microsoft Corporation
c:\windows\system32\mfc42u.dll

odbc32 6.1.7600.16385 696.00
KB (712,704 bytes) 7/13/2009 5:29 PM
Microsoft Corporation
c:\windows\system32\odbc32.dll

odbcint 6.1.7600.16385 224.00
KB (229,376 bytes) 7/13/2009 5:28 PM
Microsoft Corporation
c:\windows\system32\odbcint.dll

wmiprvse 6.1.7600.16385 360.00
KB (368,640 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\wmipr

vse.exe

cimwin32 6.1.7600.16385 1.96 MB
(2,055,168 bytes) 7/13/2009 4:48 PM
Microsoft Corporation
c:\windows\system32\wbem\cimwi

n32.dll

security 6.1.7600.16385 5.00 KB
(5,120 bytes) 7/13/2009 4:50 PM
Microsoft Corporation
c:\windows\system32\security.dll

browcli 6.1.7600.16385 57.00 KB
(58,368 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\browcli.dll

schedcli 6.1.7600.16385 23.50 KB
(24,064 bytes) 7/13/2009 4:53 PM
Microsoft Corporation
c:\windows\system32\schedcli.dll

wmi 6.1.7600.16385 5.00 KB
(5,120 bytes) 7/13/2009 5:41 PM
Microsoft Corporation
c:\windows\system32\wmi.dll

ntevt 6.1.7600.16385 260.00
KB (266,240 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\wbem\ntevt.

dll

```

```

provthrd 6.1.7600.16385 300.00
KB (307,200 bytes) 7/13/2009 4:47 PM
Microsoft Corporation
c:\windows\system32\provthrd.dll

msvcirt 7.0.7600.16385 76.50 KB
(78,336 bytes) 7/13/2009 4:18 PM
Microsoft Corporation
c:\windows\system32\msvcirt.dll

tapi32 6.1.7600.16385 243.00
KB (248,832 bytes) 7/13/2009 5:41 PM
Microsoft Corporation
c:\windows\system32\tapi32.dll

unidrvui 0.3.7600.16385 863.50
KB (884,224 bytes) 7/13/2009 6:18 PM
Microsoft Corporation
c:\windows\system32\spool\drivers
\x64\3\unidrvui.dll

mxdwdui 0.3.7600.16385 215.50
KB (220,672 bytes) 7/13/2009 6:19 PM
Microsoft Corporation
c:\windows\system32\spool\drivers
\x64\3\mxdwdui.dll

wmiperfclass 6.1.7600.16385 133.00
KB (136,192 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\wbem\wmiper

rfclass.dll

pdh 6.1.7600.16385 293.00
KB (300,032 bytes) 7/13/2009 4:31 PM
Microsoft Corporation
c:\windows\system32\pdh.dll

```

[Services]

Display Name	Name	State
Start Mode	Service Type	Path
Error Control	Start Name	Tag ID
Application Experience	AeLookupSvc	Running
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k netsvcs	Normal	LocalSystem 0
Application Layer Gateway Service	ALG	
Stopped	Manual	Own
Process	c:\windows\system32\alg.exe	
	Normal	NT
AUTHORITY\LocalService	0	
Altiris Deployment Agent	Altiris Deployment	
Agent	Running	Auto
Process	"c:\program	
files\altiris\agent\dagent.exe" -		
load=default.dll,config.dll,autoupdate.dll	Ignore	
LocalSystem	0	
Application Host Helper Service		
AppHostSvc	Running	Auto
Share Process		
	c:\windows\system32\svchost.exe -	
k apphost	Normal	LocalSystem 0
Application Identity	AppIDSvc	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k localserviceandnoimpersonation	Normal	
	NT Authority\LocalService	0
Application Information	Appinfo	Stopped
Manual	Share Process	
	c:\windows\system32\svchost.exe -	
k netsvcs	Normal	LocalSystem 0

Application Management AppMgmt Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

ASP.NET State Service aspnet_state Stopped
Manual Own Process
c:\windows\microsoft.net\framework
k64\v2.0.50727\aspnet_state.exe Normal
NT AUTHORITY\NetworkService
0

Windows Audio Endpoint Builder
AudioEndpointBuilder Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
LocalSystem 0

Windows Audio AudioSrv Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k localservicenetworkrestricted Normal
NT AUTHORITY\LocalService
0

Base Filtering Engine BFE Running
Auto Share Process
c:\windows\system32\svchost.exe -
k localservicenetwork Normal NT
AUTHORITY\LocalService 0

Background Intelligent Transfer Service BITS
Running Auto Share
Process c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Computer Browser Browser Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Certificate Propagation CertPropSvc Running
Manual Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Microsoft .NET Framework NGEN v2.0.50727_X86
clr_optimization_v2.0.50727_32
Stopped Manual Own
Process
c:\windows\microsoft.net\framework
k\v2.0.50727\mscorsvw.exe Ignore
LocalSystem 0

Microsoft .NET Framework NGEN v2.0.50727_X64
clr_optimization_v2.0.50727_64
Stopped Manual Own
Process
c:\windows\microsoft.net\framework
k64\v2.0.50727\mscorsvw.exe Ignore
LocalSystem 0

COM+ System Application COMSysApp Running
Manual Own Process
c:\windows\system32\dlhhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-
00805fc79235} Normal
LocalSystem 0

Cryptographic Services CryptSvc Running
Auto Share Process
c:\windows\system32\svchost.exe -
k networkservice Normal NT
Authority\NetworkService 0

DCOM Server Process Launcher
DcomLaunch Running Auto
Share Process
c:\windows\system32\svchost.exe -
k dcomlaunch Normal
LocalSystem 0

Disk Defragmenter defragsvc Stopped
Manual Own Process
c:\windows\system32\svchost.exe -
k defragsvc Normal localSystem 0

DHCP Client Dhcp Running Auto
Share Process
c:\windows\system32\svchost.exe -
k localservicenetworkrestricted Normal
NT Authority\LocalService 0

DNS Client Dnscache Running Auto
Share Process
c:\windows\system32\svchost.exe -
k networkservice Normal NT
AUTHORITY\NetworkService 0

Wired AutoConfig dot3svc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
localSystem 0

Diagnostic Policy Service DPS Running
Auto Share Process
c:\windows\system32\svchost.exe -
k localservicenetwork Normal NT
AUTHORITY\LocalService 0

Extensible Authentication Protocol EapHost
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k netsvcs Normal localSystem 0

Encrypting File System (EFS) EFS
Stopped Manual Share
Process c:\windows\system32\lsass.exe
Normal LocalSystem 0

Windows Event Log eventlog Running
Auto Share Process
c:\windows\system32\svchost.exe -
k localservicenetworkrestricted Normal
NT AUTHORITY\LocalService
0

COM+ Event System EventSystem Running
Auto Share Process
c:\windows\system32\svchost.exe -
k localservice Normal NT
AUTHORITY\LocalService 0

Microsoft Fibre Channel Platform Registration
Service FCRegSvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -
k localservicenetworkrestricted Normal
NT AUTHORITY\LocalService
0

Function Discovery Provider Host fdPHost
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k localservice Normal NT
AUTHORITY\LocalService 0

Function Discovery Resource Publication
FDResPub Stopped Manual
Share Process
c:\windows\system32\svchost.exe -
k localserviceandnoimpersonation Normal
NT AUTHORITY\LocalService
0

Windows Font Cache Service
FontCache Stopped Manual
Share Process
c:\windows\system32\svchost.exe -
k localserviceandnoimpersonation Normal
NT AUTHORITY\LocalService
0

Windows Presentation Foundation Font Cache
3.0.0.0 FontCache3.0.0.0 Stopped
Manual Own Process
c:\windows\microsoft.net\framework
k64\v3.0\wpf\presentationfontcache.exe Normal
NT Authority\LocalService 0

Group Policy Client gpsvc Running
Auto Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Human Interface Device Access hidserv
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
LocalSystem 0

Health Key and Certificate Management hkmsvc
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k netsvcs Normal localSystem 0

Windows CardSpace idsvc Stopped
Manual Share Process
"c:\windows\microsoft.net\framework
k64\v3.0\windows communication
foundation\infocard.exe" Normal
LocalSystem 0

IIS Admin Service IISADMIN Running
Auto Share Process
c:\windows\system32\inetmgr\inetin
fo.exe Normal LocalSystem 0

IKE and AuthIP IPsec Keying Modules IKEEXT
Running Auto Share
Process c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

PnP-X IP Bus Enumerator IPBusEnum Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
LocalSystem 0

IP Helper iphlpsvc Running Auto
Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

CNG Key Isolation KeyIso Stopped
Manual Share Process
c:\windows\system32\lsass.exe
Normal LocalSystem 0

KtmRm for Distributed Transaction Coordinator
KtmRm Stopped Manual
Share Process
c:\windows\system32\svchost.exe -
k networkserviceandnoimpersonation Normal
NT AUTHORITY\NetworkService
0

Server LanmanServer Running
Auto Share Process
c:\windows\system32\svchost.exe -
k netsvcs Normal LocalSystem 0

Workstation LanmanWorkstation Running
Auto Share Process
c:\windows\system32\svchost.exe -
k networkservice Normal NT
AUTHORITY\NetworkService 0

Link-Layer Topology Discovery Mapper lldsvc
Stopped Manual Share
Process c:\windows\system32\svchost.exe -
k localservice Normal NT
AUTHORITY\LocalService 0

TCP/IP NetBIOS Helper Imhosts Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetworkrestricted Normal
 NT AUTHORITY\LocalService
 0

Multimedia Class SchedulerMMCSS Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Windows Firewall MpsSvc Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetwork Normal NT
 Authority\LocalService 0

Distributed Transaction Coordinator MSDTC
 Running Auto Own
 Process c:\windows\system32\msdtc.exe
 Normal NT
 AUTHORITY\NetworkService 0

Microsoft iSCSI Initiator Service MSISCSI
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Windows Installer msiserver Stopped
 Manual Own Process
 c:\windows\system32\msiexec.exe
 /v Normal LocalSystem 0

SQL Server (SQLEXPRESS) MSSQL\$SQLEXPRESS
 Running Auto Own
 Process "c:\program files (x86)\microsoft
 sql server\mssql.1\mssql\bin\sqlservr.exe" -
 ssqlexpress Normal NT
 AUTHORITY\NetworkService 0

SQL Server Active Directory Helper
 MSSQLServerADHelper Stopped
 Disabled Own Process
 "c:\program files (x86)\microsoft
 sql server\90\shared\sqlad\lp90.exe" Normal
 NT AUTHORITY\NetworkService
 0

Visual Studio 2008 Remote Debugger
 msvsmon90 Stopped Disabled
 Own Process "c:\program
 files\microsoft visual studio
 9.0\common7\ide\remote
 debugger\x64\msvsmon.exe" /service
 msvsmon90 Ignore LocalSystem 0

Network Access Protection Agent
 napagent Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 AUTHORITY\NetworkService 0

Netlogon Netlogon Stopped Manual
 Share Process
 c:\windows\system32\lsass.exe
 Normal LocalSystem 0

Network Connections Netman Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localsystemnetworkrestricted Normal
 LocalSystem 0

Network List Service netprofm Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localservice Normal NT
 AUTHORITY\LocalService 0

NetTcpPortSharing Stopped
 Disabled Share Process
 "c:\windows\microsoft.net\frameworko
 rk64\v3.0\windows communication
 foundation\smsvchost.exe" Normal NT
 AUTHORITY\LocalService 0

Network Location Awareness NlaSvc
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k networkservice Normal NT
 AUTHORITY\NetworkService 0

Network Store Interface Service nsi
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k localservice Normal NT
 Authority\LocalService 0

Office Source Engine ose Stopped
 Manual Own Process
 "c:\program files (x86)\common
 files\microsoft shared\source engine\ose.exe"
 Normal LocalSystem 0

Performance Counter DLL Host PerfHost
 Stopped Manual Own
 Process c:\windows\syswow64\perfhst.exe
 Normal NT
 AUTHORITY\LocalService 0

Performance Logs & Alerts pla Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k localservicenetwork Normal NT
 AUTHORITY\LocalService 0

Plug and PlayPlugPlay Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k dcomlaunch Normal
 LocalSystem 0

IPsec Policy Agent PolicyAgent Running
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k networkservicenetworkrestricted Normal
 NT Authority\NetworkService
 0

Power Power Running Auto
 Share Process
 c:\windows\system32\svchost.exe -
 k dcomlaunch Normal
 LocalSystem 0

User Profile Service ProfSvc Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Protected Storage ProtectedStorage
 Stopped Manual Share
 Process c:\windows\system32\lsass.exe
 Normal LocalSystem 0

Remote Access Auto Connection Manager
 RasAuto Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Remote Access Connection Manager RasMan
 Stopped Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Routing and Remote Access
 RemoteAccess Stopped
 Disabled Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal localSystem 0

Remote Registry RemoteRegistry
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k regsvc Normal NT
 AUTHORITY\LocalService 0

Remote Command Service RMSYS Running
 Auto Own Process
 "c:\program files
 (x86)\benchcraft\rsys.exe" Normal
 .\Administrator 0

RPC Endpoint Mapper RpcEptMapper
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k rpcss Normal NT
 AUTHORITY\NetworkService 0

Remote Procedure Call (RPC) Locator
 RpcLocator Stopped Manual
 Own Process
 c:\windows\system32\locator.exe
 Normal NT
 AUTHORITY\NetworkService 0

Remote Procedure Call (RPC) RpcSs
 Running Auto Share
 Process c:\windows\system32\svchost.exe -
 k rpcss Normal NT
 AUTHORITY\NetworkService 0

Resultant Set of Policy Provider
 RSoPProv Stopped Manual
 Share Process
 c:\windows\system32\rsopprov.exe
 Normal LocalSystem 0

Special Administration Console Helper sacsvr
 Running Manual Share
 Process c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Security Accounts ManagerSamSs Running
 Auto Share Process
 c:\windows\system32\lsass.exe
 Normal LocalSystem 0

Smart Card SCardSvr Stopped Manual
 Share Process
 c:\windows\system32\svchost.exe -
 k localserviceandnoimpersonation Normal
 NT AUTHORITY\LocalService
 0

Task Scheduler Schedule Running
 Auto Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Smart Card Removal PolicySCPPolicySvc Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

Secondary Logon seclogon Stopped
 Manual Share Process
 c:\windows\system32\svchost.exe -
 k netsvcs Normal LocalSystem 0

System Event Notification Service	SENS	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -			
k netsvcs	Normal	LocalSystem	0	
Remote Desktop Configuration		SessionEnv	Running	Manual
Share Process	c:\windows\system32\svchost.exe -			
k netsvcs	Normal	LocalSystem	0	
Internet Connection Sharing (ICS)		SharedAccess	Stopped	
Disabled	Share Process			
Process	c:\windows\system32\svchost.exe -			
k netsvcs	Normal	LocalSystem	0	
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -			
k netsvcs	Ignore	LocalSystem	0	
SNMP Trap	SNMPTRAP	Stopped	Manual	
Own Process	c:\windows\system32\snmptrap.exe			
	Normal	NT		
AUTHORITY\LocalService	0			
Print Spooler	Spooler	Running	Auto	
Own Process	c:\windows\system32\spoolsv.exe			
	Normal	LocalSystem	0	
Software Protection	sppsvc	Stopped		
Auto	Own Process			
Process	c:\windows\system32\sppsvc.exe			
	Normal	NT		
AUTHORITY\NetworkService	0			
SPP Notification Service	sppuinotify	Stopped		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
AUTHORITY\LocalService	0			
SQL Server Browser	SQLBrowser	Stopped		
Disabled	Own Process			
"c:\program files (x86)\microsoft				
sql server\90\shared\sqlbrowser.exe"	Normal			
NT AUTHORITY\NetworkService	0			
SQL Server VSS Writer	SQLWriter	Running		
Auto	Own Process			
"c:\program files\microsoft sql				
server\90\shared\sqlwriter.exe"	Normal			
LocalSystem	0			
SSDP Discovery	SSDPSRV	Stopped		
Disabled	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localserviceandnoimpersonation	Normal			
NT AUTHORITY\LocalService	0			
Secure Socket Tunneling Protocol Service	SstpSvc	Stopped	Manual	
Share Process				
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
Authority\LocalService	0			
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual	
Own Process				
Process	c:\windows\system32\svchost.exe -			
k swprv	Normal	LocalSystem	0	

Telephony	TapiSrv	Stopped	Manual	
Own Process				
Process	c:\windows\system32\svchost.exe -			
k tapisrv	Normal	NT		
AUTHORITY\NetworkService	0			
TPM Base Services	TBS	Stopped		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localserviceandnoimpersonation	Normal			
NT AUTHORITY\LocalService	0			
Remote Desktop Services	TermService	Running		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k termsvcs	Normal	NT		
Authority\NetworkService	0			
Thread Ordering Server	THREADORDER	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
AUTHORITY\LocalService	0			
Distributed Link Tracking Client	TrkWks	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -			
k localsystemnetworkrestricted	Normal			
LocalSystem	0			
Windows Modules Installer	TrustedInstaller	Stopped	Manual	Own
Process				
c:\windows\servicing\trustedinstalle				
r.exe	Normal	LocalSystem	0	
Interactive Services Detection	UI0Detect	Stopped	Manual	
Own Process				
Process	c:\windows\system32\ui0detect.exe			
	Normal	LocalSystem	0	
Remote Desktop Services UserMode Port	UmRdpService	Running		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localsystemnetworkrestricted	Normal			
LocalSystem	0			
UPnP Device Host	upnphost	Stopped		
Disabled	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localserviceandnoimpersonation	Normal			
NT AUTHORITY\LocalService	0			
Desktop Window Manager Session Manager	UxSms	Running	Auto	
Share Process				
Process	c:\windows\system32\svchost.exe -			
k localsystemnetworkrestricted	Normal			
LocalSystem	0			
Credential Manager	VaultSvc	Stopped		
Manual	Share Process			
Process	c:\windows\system32\lsass.exe			
	Normal	LocalSystem	0	
Virtual Disk	vds	Stopped	Manual	
Own Process				
Process	c:\windows\system32\vds.exe			
	Normal	LocalSystem	0	
Volume Shadow Copy	VSS	Stopped		
Manual	Own Process			
Process	c:\windows\system32\vssvc.exe			
	Normal	LocalSystem	0	

Windows Time	W32Time	Running		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
Authority\LocalService	0			
Windows Update	WuService	Running	Auto	Share
Process	c:\windows\system32\svchost.exe -			
k iissvcs	Normal	LocalSystem	0	
Windows Process Activation Service	WAS	Running	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k iissvcs	Normal	LocalSystem	0	
Windows Color System	WcsPlugInService	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k wcssvc	Normal	NT		
AUTHORITY\LocalService	0			
Diagnostic Service Host	WdiServiceHost	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
AUTHORITY\LocalService	0			
Diagnostic System Host	WdiSystemHost	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k localsystemnetworkrestricted	Normal			
LocalSystem	0			
Windows Event Collector	Wevcsv	Stopped		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k networkservice	Normal	NT		
AUTHORITY\NetworkService	0			
Problem Reports and Solutions Control Panel	WerCplSupport	Stopped		
Support	Manual	Share Process		
Process	c:\windows\system32\svchost.exe -			
k netsvcs	Normal	LocalSystem	0	
Windows Error Reporting Service	WerSvc	Stopped	Manual	Share
Process	c:\windows\system32\svchost.exe -			
k wersvcgroup	Ignore			
LocalSystem	0			
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped		
Manual	Share Process			
Process	c:\windows\system32\svchost.exe -			
k localservice	Normal	NT		
AUTHORITY\LocalService	0			
Windows Management Instrumentation	Winmgmt	Running	Auto	
Share Process				
Process	c:\windows\system32\svchost.exe -			
k netsvcs	Ignore	LocalSystem	0	
Windows Remote Management (WS-	WinRM	Running		
Management)	Auto	Share Process		
Process	c:\windows\system32\svchost.exe -			
k networkservice	Normal	NT		
AUTHORITY\NetworkService	0			
WMI Performance Adapter	wmiApSrv	Stopped		
Manual	Own Process			
Process	c:\windows\system32\wbem\wmiapi			
srv.exe	Normal	LocalSystem	0	

```

Portable Device Enumerator Service
  WPDBusEnum      Stopped
  Manual          Share Process
  c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
  LocalSystem 0
Windows Update      wuauserv Running
  Auto             Share Process
  c:\windows\system32\svchost.exe -
k netsvcs          Normal      LocalSystem 0

Windows Driver Foundation - User-mode Driver
Framework          wuafsvc  Stopped  Manual
  Share Process
  c:\windows\system32\svchost.exe -
k localsystemnetworkrestricted Normal
  LocalSystem 0

```

[Program Groups]

Group Name	Name	User Name	Default
Start Menu	Default:Start Menu		Default
Start Menu\Programs	Default:Start Menu\Programs		Default
Start Menu\Programs\Accessories	Default:Start Menu\Programs\Accessories		Default
Start Menu\Programs\Accessories\Accessibility	Default:Start Menu\Programs\Accessories\Accessibility		Default
Start Menu\Programs\Accessories\System Tools	Default:Start Menu\Programs\Accessories\System Tools		Default
Start Menu\Programs\Accessories\System Tools	Default:Start Menu\Programs\Accessories\System Tools		Default
Start Menu\Programs\Maintenance	Default:Start Menu\Programs\Maintenance		Default
Start Menu	Public:Start Menu		Public
Start Menu\Programs	Public:Start Menu\Programs		Public
Start Menu\Programs\Accessories	Public:Start Menu\Programs\Accessories		Public
Start Menu\Programs\Accessories\Accessibility	Public:Start Menu\Programs\Accessories\Accessibility		Public
Start Menu\Programs\Accessories\System Tools	Public:Start Menu\Programs\Accessories\System Tools		Public
Start Menu\Programs\Accessories\System Tools	Public:Start Menu\Programs\Accessories\System Tools		Public
Start Menu\Programs\Accessories\Windows PowerShell	Public:Start Menu\Programs\Accessories\Windows PowerShell		Public
Start Menu\Programs\Administrative Tools	Public:Start Menu\Programs\Administrative Tools		Public
Start Menu\Programs\Administrative Tools	Public:Start Menu\Programs\Administrative Tools		Public
Start Menu\Programs\Administrative Tools\Terminal Services	Public:Start Menu\Programs\Administrative Tools\Terminal Services		Public
Start Menu\Programs\Altiris	Public:Start Menu\Programs\Altiris		Public

```

Start Menu\Programs\Altiris\Deployment Solution
  Public:Start
Menu\Programs\Altiris\Deployment Solution
Start Menu\Programs\Maintenance
  Public:Start
Menu\Programs\Maintenance      Public
Start Menu\Programs\Microsoft SQL Server 2005
  Public:Start
Menu\Programs\Microsoft SQL Server 2005
  Public
Start Menu\Programs\Microsoft SQL Server
2005\Configuration Tools      Public:Start
Menu\Programs\Microsoft SQL Server
2005\Configuration Tools      Public
Start Menu\Programs\Microsoft SQL Server 2008
  Public:Start
Menu\Programs\Microsoft SQL Server 2008
  Public
Start Menu\Programs\Microsoft SQL Server
2008\Configuration Tools      Public:Start
Menu\Programs\Microsoft SQL Server
2008\Configuration Tools      Public
Start Menu\Programs\Microsoft SQL Server 2008
R2      Public:Start
Menu\Programs\Microsoft SQL Server 2008
R2      Public
Start Menu\Programs\Microsoft SQL Server 2008
R2\Configuration Tools      Public:Start
Menu\Programs\Microsoft SQL Server 2008
R2\Configuration Tools      Public
Start Menu\Programs\Microsoft SQL Server 2008
R2\Documentation and Tutorials
  Public:Start
Menu\Programs\Microsoft SQL Server 2008
R2\Documentation and Tutorials      Public
Start Menu\Programs\Microsoft SQL Server 2008
R2\Integration Services      Public:Start
Menu\Programs\Microsoft SQL Server 2008
R2\Integration Services      Public
Start Menu\Programs\Microsoft Visual Studio
2008      Public:Start
Menu\Programs\Microsoft Visual Studio 2008
  Public
Start Menu\Programs\Microsoft Visual Studio
2008\Visual Studio Remote Tools
  Public:Start
Menu\Programs\Microsoft Visual Studio
2008\Visual Studio Remote Tools      Public
Start Menu\Programs\Microsoft Visual Studio
2008\Visual Studio Tools      Public:Start
Menu\Programs\Microsoft Visual Studio
2008\Visual Studio Tools      Public
Start Menu\Programs\Microsoft Windows SDK
v6.0A      Public:Start
Menu\Programs\Microsoft Windows SDK v6.0A
  Public
Start Menu\Programs\Microsoft Windows SDK
v6.0A\Tools      Public:Start
Menu\Programs\Microsoft Windows SDK
v6.0A\Tools      Public
Start Menu\Programs\Startup
  Public:Start Menu\Programs\Startup
  Public
Start Menu      C7KBAY2\Administrator:Start Menu
  C7KBAY2\Administrator
Start Menu\Programs
  C7KBAY2\Administrator:Start
Menu\Programs
  C7KBAY2\Administrator

```

```

Start Menu\Programs\Accessories
  C7KBAY2\Administrator:Start
Menu\Programs\Accessories
Start Menu\Programs\Accessories\Accessibility
  C7KBAY2\Administrator:Start
Menu\Programs\Accessories\Accessibility
  C7KBAY2\Administrator
Start Menu\Programs\Accessories\System Tools
  C7KBAY2\Administrator:Start
Menu\Programs\Accessories\System Tools
  C7KBAY2\Administrator
Start Menu\Programs\Administrative Tools
  C7KBAY2\Administrator:Start
Menu\Programs\Administrative Tools
  C7KBAY2\Administrator
Start Menu\Programs\Maintenance
  C7KBAY2\Administrator:Start
Menu\Programs\Maintenance
  C7KBAY2\Administrator
Start Menu\Programs\Startup
  C7KBAY2\Administrator:Start
Menu\Programs\Startup
  C7KBAY2\Administrator

```

[Startup Programs]

Program	Command	User Name	Location
DagentUI	c:\program files\altiris\client\dagentui.exe		Public
ws\CurrentVersion\Run	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run		

[OLE Registration]

Object	Local Server
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"
Paintbrush Picture	%systemroot%\system32\mspaint.exe
Package	Not Available
Microsoft PenInputPanel Control	Not Available

[Windows Error Reporting]

Time	Type	Details
5/6/2011 11:11 PM	Application Error	Faulting application name: w3wp.exe, version: 7.5.7600.16385, time stamp: 0x4a5bcd2b
Faulting module name: ole32.dll, version: 6.1.7600.16385, time stamp: 0x4a5bdac7
Exception code: 0xc0000058
Fault offset: 0x0000f771
Faulting process id: 0xb70
Faulting application start time: 0x01cc0c3ba6522bce
Faulting application path: C:\Windows\SysWOW64\inetrv\w3wp.exe
Faulting module path: C:\Windows\system32\ole32.dll
Report Id: 29883955-7836-11e0-b5f5-0026557d4c6c

5/6/2011 6:47 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0xce8
Faulting
 application start time:
 0x01cc0c1157f3a524
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: 48542640-7811-11e0-9c1f-
 0026557d4c6c

4/27/2011 8:39 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0x3948
Faulting
 application start time:
 0x01cc050f9f11e3a1
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: 7e3fd25d-710e-11e0-a065-
 0026557d4c6c

3/18/2011 9:18 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0xac48
Faulting
 application start time:
 0x01cbe5a8a5830f7a
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: 40e712b9-51a5-11e0-a4cd-
 0026557d4c6c

3/18/2011 7:58 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0x8b8
Faulting
 application start time:
 0x01cbe5a37646413a
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: 0fc67ab4-519a-11e0-ac35-
 0026557d4c6c

3/17/2011 10:25 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0xf28
Faulting
 application start time:
 0x01cbe4eb178e3fba
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: 7fdb7ee9-50e5-11e0-9fe6-
 0026557d4c6c

10/26/2014 4:26 PM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0x55c
Faulting
 application start time:
 0x01cbe40561bdb347
Faultin
 g application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: bb215136-5000-11e0-9fe6-
 0026557d4c6c

3/8/2011 5:27 AM Application Error
 Faulting application name:
 w3wp.exe, version: 7.5.7600.16385, time stamp:
 0x4a5bcd2b
Faulting module
 name: ole32.dll, version: 6.1.7600.16385, time
 stamp: 0x4a5bdac7
Exception
 code: 0xc0000005
Fault
 offset: 0x0000f771
Faulting
 process id: 0x55c
Faulting
 application start time:
 0x01cbdd1766c7a51b
Faulting
 application path:
 C:\Windows\SysWOW64\inetrv\w3wp.exe�
 0d;
Faulting module path:
 C:\Windows\syswow64\ole32.dll�
 0a;Report Id: c72a1a7a-4944-11e0-b47-
 0026557d4c6c

5/6/2011 11:11 PM Windows Error
 Reporting Fault bucket, type
 0
Event Name:
 APPCRASH
Response: Not
 available
Cab Id:
 0

Problem
 signature:
P1:
 w3wp.exe
P2:
 7.5.7600.16385
P3:
 4a5bcd2b
P4:
 ole32.dll
P5:
 6.1.7600.16385
P6:
 4a5bdac7
P7:
 c0000005
P8:
 0000f771
P9:
 
P10:
 

Attached
 files:

Thes
 e files may be available
 here:
C:\ProgramData\Micros
 oft\Windows\WER\ReportQueue\AppCrash_w3wp
 .exe_1acfc318792f45c71dad16b6277842247fc5f
 7_087fa8fc
�
 a;Analysis symbol:
 
Rechecking for solution:
 0
Report Id: 29883955-7836-
 11e0-b5f5-
 0026557d4c6c
Report Status:
 5/6/2011 11:11 PM Windows Error
 Reporting Fault bucket, type
 0
Event Name:
 APPCRASH
Response: Not
 available
Cab Id:
 0

Problem
 signature:
P1:
 w3wp.exe
P2:
 7.5.7600.16385
P3:
 4a5bcd2b
P4:
 ole32.dll
P5:
 6.1.7600.16385
P6:
 4a5bdac7
P7:
 c0000005
P8:
 0000f771
P9:
 
P10:
 

Attached
 files:

Thes
 e files may be available
 here:
C:\ProgramData\Micros
 oft\Windows\WER\ReportQueue\AppCrash_w3wp
 .exe_1acfc318792f45c71dad16b6277842247fc5f
 7_087fa8fc
�
 a;Analysis symbol:
 
Rechecking for solution:
 0
Report Id: 29883955-7836-
 11e0-b5f5-
 0026557d4c6c
Report Status:
 4

5/6/2011 6:47 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_23ff3731
�
a;Analysis symbol:

Rechecking for solution:
0
Report Id: 48542640-7811-
11e0-9c1f-
0026557d4c6c
Report Status:
0
5/6/2011 6:47 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_23ff3731
�
a;Analysis symbol:

Rechecking for solution:
0
Report Id: 48542640-7811-
11e0-9c1f-
0026557d4c6c
Report Status:
4

4/27/2011 8:39 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_0bea8759
�
0a;Analysis symbol:

Rechecking for solution:
0
Report Id: 7e3fd25d-710e-
11e0-a065-
0026557d4c6c
Report Status:
4/27/2011 8:39 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_0bea8759
�
0a;Analysis symbol:

Rechecking for solution:
0
Report Id: 7e3fd25d-710e-
11e0-a065-
0026557d4c6c
Report Status:
4

3/18/2011 9:18 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_0fb456c7
�
a;Analysis symbol:

Rechecking for solution:
0
Report Id: 40e712b9-51a5-
11e0-a4cd-
0026557d4c6c
Report Status:
3/18/2011 9:18 PM Windows Error
Reporting Fault bucket , type
0
Event Name:
APPCRAsh
Response: Not
available
Cab Id:
0

Problem
signature:
P1:
w3wp.exe
P2:
7.5.7600.16385
P3:
4a5bcd2b
P4:
ole32.dll
P5:
6.1.7600.16385
P6:
4a5bdac7
P7:
c000005
P8:
0000f71
P9:

P10:


Attached
files:

Thes
e files may be available
here:
C:\ProgramData\Micros
oft\Windows\WER\ReportQueue\AppCrash_w3wp
.exe_1acfc318792f45c71dad16b62778422247fc5f
7_0fb456c7
�
a;Analysis symbol:

Rechecking for solution:
0
Report Id: 40e712b9-51a5-
11e0-a4cd-
0026557d4c6c
Report Status:
4

3/18/2011 7:58 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_01a53716

Analysis symbol: 
Rechecking for solution: 0
Report Id: 0fc67ab4-519a-11e0-ac35-0026557d4c6c
Report Status: 0

3/18/2011 7:58 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_01a53716

Analysis symbol: 
Rechecking for solution: 0
Report Id: 0fc67ab4-519a-11e0-ac35-0026557d4c6c
Report Status: 4

3/17/2011 10:25 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_1a9dae1a

Analysis symbol: 
Rechecking for solution: 0
Report Id: 7fdb7ee9-50e5-11e0-9fe6-0026557d4c6c
Report Status: 0/17/2011 10:25 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_1a9dae1a

Analysis symbol: 
Rechecking for solution: 0
Report Id: 7fdb7ee9-50e5-11e0-9fe6-0026557d4c6c
Report Status: 4

3/16/2011 7:08 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_0f326e6d

Analysis symbol: 
Rechecking for solution: 0
Report Id: bb215136-5000-11e0-9fe6-0026557d4c6c
Report Status: 0/16/2011 7:08 PM Windows Error Reporting Fault bucket, type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f71
P9: 
P10: 

Attached files:

These files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_0f326e6d

Analysis symbol: 
Rechecking for solution: 0
Report Id: bb215136-5000-11e0-9fe6-0026557d4c6c
Report Status: 4

3/8/2011 5:27 AM Windows Error Reporting Fault bucket , type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f771
P9: 
P10: 

Attached files:

These e files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_553c322f

Analysis symbol:
Rechecking for solution: 0
Report Id: c72a1a7a-4944-11e0-b4a7-0026557d4c6c
Report Status: 0

3/8/2011 5:27 AM Windows Error Reporting Fault bucket , type 0
Event Name: APPCRASH
Response: Not available
Cab Id: 0

Problem signature:
P1: w3wp.exe
P2: 7.5.7600.16385
P3: 4a5bcd2b
P4: ole32.dll
P5: 6.1.7600.16385
P6: 4a5bdac7
P7: c000005
P8: 0000f771
P9: 
P10: 

Attached files:

These e files may be available here:
C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe_1acfc318792f45c71dad16b62778422247fc5f7_553c322f

Analysis symbol:
Rechecking for solution: 0
Report Id: c72a1a7a-4944-11e0-b4a7-0026557d4c6c
Report Status: 4

COM+ Settings

TPCC.AllTxns:
Activation:
 Enable Object Pooling selected
 Minimum Pool Size: 300
 Maximum Pool Size: 300
 Creation Timeout: 60000
 Enable Object Construction
 Enable Just In Time Activation
 Concurrency:

Concurrency Required

Microsoft IIS7 Parameters

IIS7 Manager.
 Default Web Site:
 # of connections: 150000
 CxnTimeout: 9600

Application Pool:
 DefaultAppPool:
 Advanced Settings:
 Queue Length

Queue Length: 25000

TPCC Application Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\TPCC]
"Path"="C:\inetpub\wwwroot\"
"NumberOfDeliveryThreads"=dword:00000014
"MaxConnections"=dword:000249f0
"MaxPendingDeliveries"=dword:000000c8
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="192.168.2.1,2003"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"="ssdl"
"COM_SinglePool"="YES"
```

NOTE: This is representative of 1 web client. DBServer was varied on every web client to connect to the appropriate Numa node

TCPIP Registry Parameters

No settings. Windows 2008/R2 default settings now accommodate more connections.

RTE Input Parameters

3Tier Profile

Profile: BL685cG7 82 drivers
 File Path: D:\BL685cG7\3-Tier\BL685cG7 82 drivers.xml
 Version: 5
 Number of Engines: 82
 Name: DRIVER01
 Description: Driver 1.1
 Directory: c:\tpcc\logs\driver1.log
 Machine: 15.1.120.1
 Parameter Set:
 Shutdown
 Index: 0
 Seed: 36954
 Configured Users:
 12660
 Pipe Name:
 DRIVER1358197936
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:

c:\tpcc\logs\driver2.log	Name: DRIVER02 Description: Driver 1.2 Directory: Machine: 15.1.120.1 Parameter Set: Index: 10000000 Seed: 36954 Configured Users:	12660	Configured Users: 233	233	CLIENT_NURAND: CPU: 1 Additional Options:
Shutdown		DRIVER5446286178	Pipe Name:	c:\tpcc\logs\driver9.log	Name: DRIVER09 Description: Driver 5.1 Directory:
12660		233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown	Machine: 15.1.120.5 Parameter Set:
DRIVER2362130050	Pipe Name:		CPU: 0 Additional Options:	12760	Index: 80000000 Seed: 36954 Configured Users:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver6.log	Name: DRIVER06 Description: Driver 3.2 Directory:	DRIVER1358197976	Pipe Name:
		Shutdown	Machine: 15.1.120.3 Parameter Set:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
	CPU: 1 Additional Options:		Index: 50000000 Seed: 36954 Configured Users:	233	
c:\tpcc\logs\driver3.log	Name: DRIVER03 Description: Driver 2.1 Directory:	12660	Pipe Name:		CPU: 0 Additional Options:
Shutdown	Machine: 15.1.120.2 Parameter Set:	DRIVER6446398233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver10.log	Name: DRIVER10 Description: Driver 5.2 Directory:
12660	Index: 20000000 Seed: 36954 Configured Users:	233	CPU: 1 Additional Options:	Shutdown	Machine: 15.1.120.5 Parameter Set:
DRIVER3362233588	Pipe Name:		Name: DRIVER07 Description: Driver 4.1 Directory:	12760	Index: 90000000 Seed: 36954 Configured Users:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver7.log	Machine: 15.1.120.4 Parameter Set:	DRIVER2362137050	Pipe Name:
		Shutdown	Index: 60000000 Seed: 36954 Configured Users:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
	CPU: 0 Additional Options:		Pipe Name:	233	
c:\tpcc\logs\driver4.log	Name: DRIVER04 Description: Driver 2.2 Directory:	12660	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver11.log	CPU: 1 Additional Options:
Shutdown	Machine: 15.1.120.2 Parameter Set:	DRIVER7446448887	CPU: 0 Additional Options:	Shutdown	Name: DRIVER11 Description: Driver 6.1 Directory:
12660	Index: 30000000 Seed: 36954 Configured Users:	233	Name: DRIVER08 Description: Driver 4.2 Directory:		Machine: 15.1.120.6 Parameter Set:
DRIVER4362250296	Pipe Name:		Machine: 15.1.120.4 Parameter Set:	12760	Index: 100000000 Seed: 36954 Configured Users:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver8.log	Index: 70000000 Seed: 36954 Configured Users:	DRIVER3362237588	Pipe Name:
		Shutdown	Pipe Name:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
	CPU: 1 Additional Options:		Name: DRIVER05 Description: Driver 3.1 Directory:	233	
c:\tpcc\logs\driver5.log	Name: DRIVER05 Description: Driver 3.1 Directory:	12660	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10	c:\tpcc\logs\driver12.log	CPU: 0 Additional Options:
Shutdown	Machine: 15.1.120.3 Parameter Set:	DRIVER8446479042	CPU: 0 Additional Options:		Name: DRIVER12 Description: Driver 6.2 Directory:
	Index: 40000000		Pipe Name:		Machine: 15.1.120.6

Shutdown	Parameter Set: Index: 110000000 Seed: 36954 Configured Users:	233	Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND: CPU: 0 Additional Options:	10.1 c:\tpcc\logs\driver19.log	Description: Driver Directory: Machine: 15.1.120.10 Parameter Set:
12760 DRIVER4362257296	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER16 Description: Driver 8.2 Directory: Machine: 15.1.120.8 Parameter Set:	Shutdown	Index: 180000000 Seed: 36954 Configured Users:
233	CPU: 1 Additional Options: Name: DRIVER13 Description: Driver 7.1 Directory: Machine: 15.1.120.7 Parameter Set:	c:\tpcc\logs\driver16.log	Index: 150000000 Seed: 36954 Configured Users:	11950 DRIVER3352233588	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver13.log	CPU: 0 Additional Options: Name: DRIVER14 Description: Driver 7.2 Directory: Machine: 15.1.120.7 Parameter Set:	Shutdown	Index: 160000000 Seed: 36954 Configured Users:	233	CPU: 0 Additional Options: Name: DRIVER20 Description: Driver
Shutdown	Index: 120000000 Seed: 36954 Configured Users:	233	CPU: 1 Additional Options: Name: DRIVER17 Description: Driver 9.1 Directory: Machine: 15.1.120.9 Parameter Set:	10.2 c:\tpcc\logs\driver20.log	Directory: Machine: 15.1.120.10 Parameter Set:
12760 DRIVER5446287178	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver17.log	Index: 160000000 Seed: 36954 Configured Users:	Shutdown	Index: 190000000 Seed: 36954 Configured Users:
233	CPU: 0 Additional Options: Name: DRIVER14 Description: Driver 7.2 Directory: Machine: 15.1.120.7 Parameter Set:	Shutdown	Index: 160000000 Seed: 36954 Configured Users:	11950 DRIVER4352250296	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver14.log	CPU: 1 Additional Options: Name: DRIVER15 Description: Driver 8.1 Directory: Machine: 15.1.120.8 Parameter Set:	12760 DRIVER1458197936	CPU: 0 Additional Options: Name: DRIVER18 Description: Driver 9.2 Directory: Machine: 15.1.120.9 Parameter Set:	233	CPU: 1 Additional Options: Name: DRIVER21 Description: Driver
Shutdown	Index: 130000000 Seed: 36954 Configured Users:	233	CPU: 0 Additional Options: Name: DRIVER18 Description: Driver 9.2 Directory: Machine: 15.1.120.9 Parameter Set:	11.1 c:\tpcc\logs\driver21.log	Directory: Machine: 15.1.120.11 Parameter Set:
12760 DRIVER6446397233	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver18.log	Index: 170000000 Seed: 36954 Configured Users:	Shutdown	Index: 200000000 Seed: 36954 Configured Users:
233	CPU: 1 Additional Options: Name: DRIVER15 Description: Driver 8.1 Directory: Machine: 15.1.120.8 Parameter Set:	Shutdown	Index: 170000000 Seed: 36954 Configured Users:	11950 DRIVER5456286178	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver15.log	CPU: 0 Additional Options: Name: DRIVER15 Description: Driver 8.1 Directory: Machine: 15.1.120.8 Parameter Set:	11950 DRIVER2462130050	CPU: 0 Additional Options: Name: DRIVER18 Description: Driver 9.2 Directory: Machine: 15.1.120.9 Parameter Set:	233	CPU: 0 Additional Options: Name: DRIVER22 Description: Driver
Shutdown	Index: 140000000 Seed: 36954 Configured Users:	233	CPU: 1 Additional Options: Name: DRIVER19	11.2 c:\tpcc\logs\driver22.log	Directory: Machine: 15.1.120.11 Parameter Set:
12760 DRIVER7446447887	Pipe Name: Connect Rate: 10000			Shutdown	

11950	Index: 210000000 Seed: 36954 Configured Users:	233	Start Rate: 9000000 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	15.1	Additional Options: Name: DRIVER29 Description: Driver
DRIVER6456398233	Pipe Name:		CPU: 0 Additional Options:	c:\tpcc\logs\driver29.log	Directory:
	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	13.2	Name: DRIVER26 Description: Driver	Shutdown	Machine: 15.1.120.15 Parameter Set:
233	CPU: 1 Additional Options:	c:\tpcc\logs\driver26.log	Directory:	11950	Index: 280000000 Seed: 36954 Configured Users:
	Name: DRIVER23 Description: Driver	Shutdown	Machine: 15.1.120.13 Parameter Set:	DRIVER5456287178	Pipe Name:
12.1	Directory:		Index: 250000000 Seed: 36954 Configured Users:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver23.log	Machine: 15.1.120.12 Parameter Set:	11950	Pipe Name:	233	CPU: 0 Additional Options:
Shutdown	Index: 220000000 Seed: 36954 Configured Users:	DRIVER2352137050	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER30 Description: Driver
11950	Pipe Name:	233	CPU: 1 Additional Options:	15.2	Directory:
DRIVER7456448887	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER27 Description: Driver	c:\tpcc\logs\driver30.log	Machine: 15.1.120.15 Parameter Set:
233	CPU: 0 Additional Options:	14.1	Directory:	Shutdown	Index: 290000000 Seed: 36954 Configured Users:
	Name: DRIVER24 Description: Driver	c:\tpcc\logs\driver27.log	Machine: 15.1.120.14 Parameter Set:	11950	Pipe Name:
12.2	Directory:	Shutdown	Index: 260000000 Seed: 36954 Configured Users:	DRIVER6456397233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver24.log	Machine: 15.1.120.12 Parameter Set:	11950	Pipe Name:	233	CPU: 1 Additional Options:
Shutdown	Index: 230000000 Seed: 36954 Configured Users:	DRIVER3352237588	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER31 Description: Driver
11950	Pipe Name:	233	CPU: 0 Additional Options:	16.1	Directory:
DRIVER8456479042	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER28 Description: Driver	c:\tpcc\logs\driver31.log	Machine: 15.1.120.16 Parameter Set:
233	CPU: 1 Additional Options:	14.2	Directory:	Shutdown	Index: 300000000 Seed: 36954 Configured Users:
	Name: DRIVER25 Description: Driver	c:\tpcc\logs\driver28.log	Machine: 15.1.120.14 Parameter Set:	11950	Pipe Name:
13.1	Directory:	Shutdown	Index: 270000000 Seed: 36954 Configured Users:	DRIVER7456447887	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver25.log	Machine: 15.1.120.13 Parameter Set:	11950	Pipe Name:	233	CPU: 0 Additional Options:
Shutdown	Index: 240000000 Seed: 36954 Configured Users:	DRIVER4352257296	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:		Name: DRIVER32 Description: Driver
11950	Pipe Name:	233	CPU: 1	16.2	Directory:
DRIVER1348197976				c:\tpcc\logs\driver32.log	

Shutdown	Machine: 15.1.120.16 Parameter Set:	11950 DRIVER3363233588	Configured Users: Pipe Name:	233	CLIENT_NURAND: 10 CPU: 1 Additional Options:
11950	Index: 310000000 Seed: 36954 Configured Users:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	20.1	Name: DRIVER39 Description: Driver
DRIVER8453477042	Pipe Name:	233	CPU: 0 Additional Options:	c:\tpcc\logs\driver39.log	Directory:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	18.2	Name: DRIVER36 Description: Driver	Shutdown	Machine: 15.1.120.38 Parameter Set:
17.1	CPU: 1 Additional Options:	c:\tpcc\logs\driver36.log	Directory:	11950	Index: 380000000 Seed: 36954 Configured Users:
c:\tpcc\logs\driver33.log	Name: DRIVER33 Description: Driver	Shutdown	Machine: 15.1.120.18 Parameter Set:	DRIVER7443448887	Pipe Name:
Shutdown	Directory:	11950	Index: 350000000 Seed: 36954 Configured Users:	233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
11950	Machine: 15.1.120.17 Parameter Set:	DRIVER4363250296	Pipe Name:	233	CPU: 0 Additional Options:
DRIVER1353197936	Index: 320000000 Seed: 36954 Configured Users:	233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	20.2	Name: DRIVER40 Description: Driver
233	Pipe Name:	19.1	CPU: 1 Additional Options:	c:\tpcc\logs\driver40.log	Directory:
17.2	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver37.log	Name: DRIVER37 Description: Driver	Shutdown	Machine: 15.1.120.19 Parameter Set:
c:\tpcc\logs\driver34.log	CPU: 0 Additional Options:	Shutdown	Directory:	11950	Index: 390000000 Seed: 36954 Configured Users:
Shutdown	Name: DRIVER34 Description: Driver	11950	Machine: 15.1.120.37 Parameter Set:	DRIVER8443479042	Pipe Name:
11950	Directory:	DRIVER5443286178	Index: 360000000 Seed: 36954 Configured Users:	233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
DRIVER2363130050	Machine: 15.1.120.17 Parameter Set:	233	Pipe Name:	21.1	CPU: 1 Additional Options:
233	Index: 330000000 Seed: 36954 Configured Users:	19.2	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver41.log	Name: DRIVER41 Description: Driver
18.1	Pipe Name:	c:\tpcc\logs\driver38.log	CPU: 0 Additional Options:	Shutdown	Directory:
c:\tpcc\logs\driver35.log	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown	Name: DRIVER38 Description: Driver	11950	Machine: 15.1.120.19 Parameter Set:
Shutdown	CPU: 1 Additional Options:	11950	Directory:	DRIVER1353197976	Index: 400000000 Seed: 36954 Configured Users:
233	Name: DRIVER35 Description: Driver	DRIVER6443398233	Machine: 15.1.120.37 Parameter Set:	233	Pipe Name:
11950	Directory:		Index: 370000000 Seed: 36954 Configured Users:		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
Shutdown	Machine: 15.1.120.18 Parameter Set:		Pipe Name:		CPU: 0 Additional Options:
233	Index: 340000000 Seed: 36954		Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1		Name: DRIVER42

21.2	Description: Driver	Shutdown	Parameter Set: Index: 440000000 Seed: 36954 Configured Users:	DRIVER8343477042	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver42.log	Directory:	11950		233	CPU: 1 Additional Options:
Shutdown	Machine: 15.1.120.20 Parameter Set:	DRIVER5443287178	Pipe Name:	25.1	Name: DRIVER49 Description: Driver
11950	Index: 410000000 Seed: 36954 Configured Users:	233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver49.log	Directory: Machine: 15.1.120.23 Parameter Set:
DRIVER2363137050	Pipe Name:	23.2	CPU: 0 Additional Options:	Shutdown	Index: 480000000 Seed: 36954 Configured Users:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver46.log	Name: DRIVER46 Description: Driver	11950	Pipe Name:
22.1	CPU: 1 Additional Options:	Shutdown	Directory:	DRIVER1453197936	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver43.log	Name: DRIVER43 Description: Driver	11950	Machine: 15.1.120.22 Parameter Set:	233	CPU: 0 Additional Options:
Shutdown	Directory:	DRIVER6443397233	Index: 450000000 Seed: 36954 Configured Users:	25.2	Name: DRIVER50 Description: Driver
11950	Machine: 15.1.120.20 Parameter Set:	233	Pipe Name:	c:\tpcc\logs\driver50.log	Directory: Machine: 15.1.120.24 Parameter Set:
DRIVER3363237588	Index: 420000000 Seed: 36954 Configured Users:	24.1	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown	Index: 490000000 Seed: 36954 Configured Users:
233	Pipe Name:	c:\tpcc\logs\driver47.log	CPU: 1 Additional Options:	11950	Pipe Name:
22.2	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown	Name: DRIVER47 Description: Driver	DRIVER2463130050	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver44.log	CPU: 0 Additional Options:	11950	Directory:	233	CPU: 1 Additional Options:
Shutdown	Name: DRIVER44 Description: Driver	DRIVER7443447887	Machine: 15.1.120.22 Parameter Set:	26.1	Name: DRIVER51 Description: Driver
11950	Directory:	233	Index: 460000000 Seed: 36954 Configured Users:	c:\tpcc\logs\driver51.log	Directory: Machine: 15.1.120.24 Parameter Set:
DRIVER4363257296	Machine: 15.1.120.21 Parameter Set:	24.2	Pipe Name:	Shutdown	Index: 500000000 Seed: 36954 Configured Users:
233	Index: 430000000 Seed: 36954 Configured Users:	c:\tpcc\logs\driver48.log	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	11950	Pipe Name:
23.1	Pipe Name:	Shutdown	CPU: 0 Additional Options:	DRIVER3353233588	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10
c:\tpcc\logs\driver45.log	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	11950	Name: DRIVER48 Description: Driver		
	CPU: 1 Additional Options:		Directory:		
	Name: DRIVER45 Description: Driver		Machine: 15.1.120.23 Parameter Set:		
	Directory:		Index: 470000000 Seed: 36954 Configured Users:		
	Machine: 15.1.120.21				

233	CLIENT_NURAND: CPU: 0 Additional Options: Name: DRIVER52 Description: Driver	28.1 c:\tpcc\logs\driver55.log Shutdown	Description: Driver Directory: Machine: 15.1.120.26 Parameter Set: Index: 540000000 Seed: 36954 Configured Users:	11950 DRIVER2353137050	Seed: 36954 Index: 540000000 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
26.2	Directory: Machine: 15.1.120.25 Parameter Set: Index: 510000000 Seed: 36954 Configured Users:	11950 DRIVER7453448887	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	233	CPU: 1 Additional Options: Name: DRIVER59 Description: Driver
c:\tpcc\logs\driver52.log	Shutdown	233	CPU: 0 Additional Options: Name: DRIVER56 Description: Driver	c:\tpcc\logs\driver59.log	Directory: Machine: 15.1.120.28 Parameter Set: Index: 580000000 Seed: 36954 Configured Users:
11950	Index: 510000000 Seed: 36954 Configured Users:	233	Directory: Machine: 15.1.120.27 Parameter Set: Index: 550000000 Seed: 36954 Configured Users:	30.1	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
DRIVER4353250296	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	28.2	Directory: Machine: 15.1.120.27 Parameter Set: Index: 550000000 Seed: 36954 Configured Users:	c:\tpcc\logs\driver59.log	Machine: 15.1.120.28 Parameter Set: Index: 580000000 Seed: 36954 Configured Users:
233	CPU: 1 Additional Options: Name: DRIVER53 Description: Driver	c:\tpcc\logs\driver56.log Shutdown	Index: 550000000 Seed: 36954 Configured Users:	Shutdown	Index: 580000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
27.1	Directory: Machine: 15.1.120.25 Parameter Set: Index: 520000000 Seed: 36954 Configured Users:	11950 DRIVER8453479042	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	11950	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver53.log	Shutdown	233	CPU: 1 Additional Options: Name: DRIVER57 Description: Driver	233	CPU: 0 Additional Options: Name: DRIVER60 Description: Driver
11950	Index: 520000000 Seed: 36954 Configured Users:	233	Directory: Machine: 15.1.120.27 Parameter Set: Index: 560000000 Seed: 36954 Configured Users:	30.2	Directory: Machine: 15.1.120.29 Parameter Set: Index: 590000000 Seed: 36954 Configured Users:
DRIVER5453286178	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	29.1	Directory: Machine: 15.1.120.27 Parameter Set: Index: 560000000 Seed: 36954 Configured Users:	c:\tpcc\logs\driver60.log	Machine: 15.1.120.29 Parameter Set: Index: 590000000 Seed: 36954 Configured Users:
233	CPU: 0 Additional Options: Name: DRIVER54 Description: Driver	c:\tpcc\logs\driver57.log Shutdown	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown	Index: 590000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
27.2	Directory: Machine: 15.1.120.26 Parameter Set: Index: 530000000 Seed: 36954 Configured Users:	11950 DRIVER1343197976	Index: 560000000 Seed: 36954 Configured Users:	11950	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
c:\tpcc\logs\driver54.log	Shutdown	233	Directory: Machine: 15.1.120.28 Parameter Set: Index: 600000000 Seed: 36954 Configured Users:	DRIVER4353257296	Machine: 15.1.120.29 Parameter Set: Index: 600000000 Seed: 36954 Configured Users:
11950	Index: 530000000 Seed: 36954 Configured Users:	233	Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	DRIVER4353257296	CPU: 1 Additional Options: Name: DRIVER61 Description: Driver
DRIVER6453398233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	c:\tpcc\logs\driver58.log Shutdown	CPU: 0 Additional Options: Name: DRIVER58 Description: Driver	31.1	Directory: Machine: 15.1.120.29 Parameter Set: Index: 600000000 Seed: 36954 Configured Users:
233	CPU: 1 Additional Options: Name: DRIVER55	29.2	Directory: Machine: 15.1.120.28 Parameter Set:	Shutdown	Index: 600000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000
c:\tpcc\logs\driver55.log	Shutdown	c:\tpcc\logs\driver58.log	Machine: 15.1.120.28 Parameter Set:	11950	Pipe Name: Connect Rate: 10000
233	CPU: 1 Additional Options: Name: DRIVER55	Shutdown	Machine: 15.1.120.28 Parameter Set:	DRIVER5453287178	Pipe Name: Connect Rate: 10000

233	Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND: CPU: 0 Additional Options:	33.1 c:\tpcc\logs\driver65.log	Additional Options: Name: DRIVER65 Description: Driver Directory: Machine: 15.1.120.31 Parameter Set:	Shutdown 11890 DRIVER6883959796	Parameter Set: 120.33 Index: 670000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
31.2	Name: DRIVER62 Description: Driver Directory: Machine: 15.1.120.30 Parameter Set:	Shutdown 11890 DRIVER6583520281	Index: 640000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	233 CPU: 1 Additional Options: Name: DRIVER69 Description: Driver Directory: Machine: 15.1.120.33 Parameter Set:	CPU: 1 Additional Options: Name: DRIVER69 Description: Driver Directory: Machine: 15.1.120.33 Parameter Set:
Shutdown	Index: 610000000 Seed: 36954 Configured Users: Pipe Name:	233 CPU: 0 Additional Options:	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	35.1 c:\tpcc\logs\driver69.log	CPU: 1 Additional Options: Name: DRIVER69 Description: Driver Directory: Machine: 15.1.120.33 Parameter Set:
11890	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	33.2 c:\tpcc\logs\driver66.log	Additional Options: Name: DRIVER66 Description: Driver Directory: Machine: 15.1.120.32 Parameter Set:	Shutdown 11890 DRIVER69169036484	Index: 680000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
233	CPU: 1 Additional Options: Name: DRIVER63 Description: Driver Directory: Machine: 15.1.120.30 Parameter Set:	Shutdown 11890 DRIVER6683747718	Index: 650000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	233 CPU: 1 Additional Options:	CPU: 0 Additional Options: Name: DRIVER70 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
32.1	Name: DRIVER63 Description: Driver Directory: Machine: 15.1.120.30 Parameter Set:	233 CPU: 1 Additional Options:	Index: 620000000 Seed: 36954 Configured Users: Pipe Name:	35.2 c:\tpcc\logs\driver70.log	CPU: 0 Additional Options: Name: DRIVER70 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
Shutdown	Index: 620000000 Seed: 36954 Configured Users: Pipe Name:	34.1 c:\tpcc\logs\driver67.log	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown 11890 DRIVER70169122015	Index: 690000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:
11890	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	233 CPU: 0 Additional Options:	Index: 660000000 Seed: 36954 Configured Users: Pipe Name:	233 CPU: 1 Additional Options:	CPU: 1 Additional Options: Name: DRIVER71 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
233	Name: DRIVER64 Description: Driver Directory: Machine: 15.1.120.31 Parameter Set:	Shutdown 11890 DRIVER6783894281	Index: 660000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	36.1 c:\tpcc\logs\driver71.log	CPU: 1 Additional Options: Name: DRIVER71 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
Shutdown	Index: 630000000 Seed: 36954 Configured Users: Pipe Name:	233 CPU: 0 Additional Options:	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	Shutdown 11890 DRIVER70169122015	CPU: 1 Additional Options: Name: DRIVER71 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
11890	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	34.2 c:\tpcc\logs\driver68.log	Index: 630000000 Seed: 36954 Configured Users: Pipe Name: Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	233 CPU: 1 Additional Options:	CPU: 1 Additional Options: Name: DRIVER71 Description: Driver Directory: Machine: 15.1.120.34 Parameter Set:
233	CPU: 1 Additional Options:	34.2 c:\tpcc\logs\driver68.log	Index: 700000000 Seed: 36954 Configured Users:	11890 DRIVER70169122015	Index: 700000000 Seed: 36954 Configured Users:

DRIVER71169231390
 Pipe Name:
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:
 Name: DRIVER72
 Description: Driver
 36.2
 Directory:
 c:\tpcc\logs\driver72.log
 Machine: 15.1.120.35
 Parameter Set:
 Shutdown
 Index: 710000000
 Seed: 36954
 Configured Users:
 11890
 Pipe Name:
 DRIVER72169313625
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 1
 Additional Options:
 Name: DRIVER73
 Description: Driver
 32.1
 Directory:
 c:\tpcc\logs\driver73.log
 Machine: 15.1.120.35
 Parameter Set:
 Shutdown
 Index: 720000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER74534478872
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:
 Name: DRIVER74
 Description: Driver
 32.2
 Directory:
 c:\tpcc\logs\driver74.log
 Machine: 15.1.120.36
 Parameter Set:
 Shutdown
 Index: 730000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER84534770432
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10

233
 CLIENT_NURAND:
 CPU: 1
 Additional Options:
 Name: DRIVER75
 Description: Driver
 33.1
 Directory:
 c:\tpcc\logs\driver75.log
 Machine: 15.1.120.36
 Parameter Set:
 Shutdown
 Index: 740000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER65835202812
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:
 Name: DRIVER76
 Description: Driver
 33.2
 Directory:
 c:\tpcc\logs\driver76.log
 Machine: 15.1.120.39
 Parameter Set:
 Shutdown
 Index: 750000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER66837477182
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 1
 Additional Options:
 Name: DRIVER77
 Description: Driver
 34.1
 Directory:
 c:\tpcc\logs\driver77.log
 Machine: 15.1.120.39
 Parameter Set:
 Shutdown
 Index: 760000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER67838942812
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:
 Name: DRIVER78

34.2
 Description: Driver
 Directory:
 c:\tpcc\logs\driver78.log
 Machine: 15.1.120.40
 Parameter Set:
 Shutdown
 Index: 770000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER68839597962
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 1
 Additional Options:
 Name: DRIVER79
 Description: Driver
 35.1
 Directory:
 c:\tpcc\logs\driver79.log
 Machine: 15.1.120.40
 Parameter Set:
 Shutdown
 Index: 780000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER691690364842
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 0
 Additional Options:
 Name: DRIVER80
 Description: Driver
 35.2
 Directory:
 c:\tpcc\logs\driver80.log
 Machine: 15.1.120.41
 Parameter Set:
 Shutdown
 Index: 790000000
 Seed: 36954
 Configured Users:
 12650
 Pipe Name:
 DRIVER701691220152
 Connect Rate: 10000
 Start Rate: 900
 Max. Concurrency: -1
 Concurrency Rate: 10
 CLIENT_NURAND:
 233
 CPU: 1
 Additional Options:
 Name: DRIVER81
 Description: Driver
 36.1
 Directory:
 c:\tpcc\logs\driver81.log
 Machine: 15.1.120.41
 Parameter Set:
 Shutdown

12650	Index: 80000000 Seed: 36954 Configured Users:	DRIVER03	Scale Down: No Driver Engine:	DRIVER07	Scale Down: No Driver Engine:
DRIVER711692313902	Pipe Name:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:
233	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	3798	Database: tpcc User: sa Protocol: HTML w_id Range: 2533 -	8862	Database: tpcc User: sa Protocol: HTML w_id Range: 7597 -
36.2	CPU: 0 Additional Options:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
c:\tpcc\logs\driver82.log	Name: DRIVER82 Description: Driver	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
Shutdown	Directory:		Scale: Normal User Count: 12660 District id: 1 Scale Down: No		Scale: Normal User Count: 12660 District id: 1 Scale Down: No
12650	Machine: 15.1.120.38 Parameter Set:	DRIVER04	Driver Engine:	DRIVER08	Driver Engine:
DRIVER721693136252	Index: 810000000 Seed: 36954 Configured Users:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:
233	Pipe Name:	5064	Database: tpcc User: sa Protocol: HTML w_id Range: 3799 -	10128	Database: tpcc User: sa Protocol: HTML w_id Range: 8863 -
	Connect Rate: 10000 Start Rate: 900 Max. Concurrency: -1 Concurrency Rate: 10 CLIENT_NURAND:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
	CPU: 1 Additional Options:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Number of User groups: 82		Scale: Normal User Count: 12660 District id: 1 Scale Down: No		Scale: Normal User Count: 12660 District id: 1 Scale Down: No
DRIVER01	Driver Engine:	DRIVER05	Driver Engine:	DRIVER09	Driver Engine:
sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.3 SQL Server:
1	Database: tpcc User: sa Protocol: HTML w_id Range: 1 - 1266 w_id Min Warehouse:	6330	Database: tpcc User: sa Protocol: HTML w_id Range: 5065 -	11404	Database: tpcc User: sa Protocol: HTML w_id Range: 10129 -
99921	w_id Max Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
	Scale: Normal User Count: 12660 District id: 1 Scale Down: No	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
DRIVER02	Driver Engine:	DRIVER06	Driver Engine:	DRIVER10	Driver Engine:
sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.2 SQL Server:	sqloctanec	IIS Server: 15.1.115.3 SQL Server:
2532	Database: tpcc User: sa Protocol: HTML w_id Range: 1267 -	7596	Database: tpcc User: sa Protocol: HTML w_id Range: 6331 -	12680	Database: tpcc User: sa Protocol: HTML w_id Range: 11405 -
1	w_id Min Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
99921	w_id Max Warehouse:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Scale: Normal User Count: 12660 District id: 1		Scale: Normal User Count: 12660 District id: 1		Scale: Normal User Count: 12760 District id: 1

DRIVER11	Scale Down: No Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 12681 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER15	Scale Down: No Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 17785 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER19	Scale Down: No Driver Engine: IIS Server: 15.1.115.4 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 22808 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No
sqloctanec	13956 1 99921	sqloctanec	19060 1 99921	sqloctanec	24002 1 99921
DRIVER12	Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 13957 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER16	Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 19061 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER20	Driver Engine: IIS Server: 15.1.115.4 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 24003 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No
sqloctanec	15232 1 99921	sqloctanec	20336 1 99921	sqloctanec	25197 1 99921
DRIVER13	Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 15233 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER17	Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 20337 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760 District id: 1 Scale Down: No	DRIVER21	Driver Engine: IIS Server: 15.1.115.4 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 25198 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No
sqloctanec	16508 1 99921	sqloctanec	21612 1 99921	sqloctanec	26392 1 99921
DRIVER14	Driver Engine: IIS Server: 15.1.115.3 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 16509 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 12760	DRIVER18	Driver Engine: IIS Server: 15.1.115.4 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 21613 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11950	DRIVER22	Driver Engine: IIS Server: 15.1.115.4 SQL Server: Database: tpcc User: sa Protocol: HTML w_id Range: 26393 - w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11950
sqloctanec	17784 1 99921	sqloctanec	22807 1 99921	sqloctanec	27587 1 99921

DRIVER23	District id: 1 Scale Down: No	DRIVER27	District id: 11950 Scale Down: No	DRIVER31	District id: 11950 Scale Down: No
sqloctanec	Driver Engine: IIS Server: 15.1.115.4 SQL Server:	sqloctanec	Driver Engine: IIS Server: 15.1.115.4 SQL Server:	sqloctanec	Driver Engine: IIS Server: 15.1.115.5 SQL Server:
28782	Database: tpcc User: sa Protocol: HTML w_id Range: 27588 -	33562	Database: tpcc User: sa Protocol: HTML w_id Range: 32368 -	38342	Database: tpcc User: sa Protocol: HTML w_id Range: 37148 -
1	w_id Min Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
99921	w_id Max Warehouse:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER24	Driver Engine:	DRIVER28	Driver Engine:	DRIVER32	Driver Engine:
sqloctanec	IIS Server: 15.1.115.4 SQL Server:	sqloctanec	IIS Server: 15.1.115.4 SQL Server:	sqloctanec	IIS Server: 15.1.115.5 SQL Server:
29977	Database: tpcc User: sa Protocol: HTML w_id Range: 28783 -	34757	Database: tpcc User: sa Protocol: HTML w_id Range: 33563 -	39537	Database: tpcc User: sa Protocol: HTML w_id Range: 38343 -
1	w_id Min Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
99921	w_id Max Warehouse:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER25	Driver Engine:	DRIVER29	Driver Engine:	DRIVER33	Driver Engine:
sqloctanec	IIS Server: 15.1.115.4 SQL Server:	sqloctanec	IIS Server: 15.1.115.5 SQL Server:	sqloctanec	IIS Server: 15.1.115.5 SQL Server:
31172	Database: tpcc User: sa Protocol: HTML w_id Range: 29978 -	35952	Database: tpcc User: sa Protocol: HTML w_id Range: 34758 -	40732	Database: tpcc User: sa Protocol: HTML w_id Range: 39538 -
1	w_id Min Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
99921	w_id Max Warehouse:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No		Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER26	Driver Engine:	DRIVER30	Driver Engine:	DRIVER34	Driver Engine:
sqloctanec	IIS Server: 15.1.115.4 SQL Server:	sqloctanec	IIS Server: 15.1.115.5 SQL Server:	sqloctanec	IIS Server: 15.1.115.5 SQL Server:
32367	Database: tpcc User: sa Protocol: HTML w_id Range: 31173 -	37147	Database: tpcc User: sa Protocol: HTML w_id Range: 35953 -	41927	Database: tpcc User: sa Protocol: HTML w_id Range: 40733 -
1	w_id Min Warehouse:	1	w_id Min Warehouse:	1	w_id Min Warehouse:
99921	w_id Max Warehouse:	99921	w_id Max Warehouse:	99921	w_id Max Warehouse:
	Scale: Normal		Scale: Normal		Scale: Normal

DRIVER35	User Count: 11950 District id: 1 Scale Down: No Driver Engine: IIS Server: 15.1.115.5 SQL Server:	DRIVER39	User Count: 11950 District id: 1 Scale Down: No Driver Engine: IIS Server: 15.1.115.5 SQL Server:	DRIVER43	User Count: 11950 District id: 1 Scale Down: No Driver Engine: IIS Server: 15.1.115.6 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 41928 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 46708 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 51488 -
43122	w_id Min Warehouse:	47902	w_id Min Warehouse:	52682	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER36	Driver Engine: IIS Server: 15.1.115.5 SQL Server:	DRIVER40	Driver Engine: IIS Server: 15.1.115.6 SQL Server:	DRIVER44	Driver Engine: IIS Server: 15.1.115.6 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 43123 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 47903 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 52683 -
44317	w_id Min Warehouse:	49097	w_id Min Warehouse:	53877	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER37	Driver Engine: IIS Server: 15.1.115.5 SQL Server:	DRIVER41	Driver Engine: IIS Server: 15.1.115.6 SQL Server:	DRIVER45	Driver Engine: IIS Server: 15.1.115.6 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 44318 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 49098 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 53878 -
45512	w_id Min Warehouse:	50292	w_id Min Warehouse:	55072	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER38	Driver Engine: IIS Server: 15.1.115.5 SQL Server:	DRIVER42	Driver Engine: IIS Server: 15.1.115.6 SQL Server:	DRIVER46	Driver Engine: IIS Server: 15.1.115.6 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 45513 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 50293 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 55073 -
46707	w_id Min Warehouse:	51487	w_id Min Warehouse:	56267	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No

	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER47	Driver Engine:		Driver Engine:		Driver Engine:
	IIS Server: 15.1.115.6 SQL Server:	DRIVER51	IIS Server: 15.1.115.7 SQL Server:	DRIVER55	IIS Server: 15.1.115.7 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 56268 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 61048 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 65828 -
57462	w_id Min Warehouse:	62242	w_id Min Warehouse:	67022	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER48	Driver Engine:		Driver Engine:		Driver Engine:
	IIS Server: 15.1.115.6 SQL Server:	DRIVER52	IIS Server: 15.1.115.7 SQL Server:	DRIVER56	IIS Server: 15.1.115.7 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 57463 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 62243 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 67023 -
58657	w_id Min Warehouse:	63437	w_id Min Warehouse:	68217	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER49	Driver Engine:		Driver Engine:		Driver Engine:
	IIS Server: 15.1.115.6 SQL Server:	DRIVER53	IIS Server: 15.1.115.7 SQL Server:	DRIVER57	IIS Server: 15.1.115.7 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 58658 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 63438 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 68218 -
59852	w_id Min Warehouse:	64632	w_id Min Warehouse:	69412	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No	99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No
DRIVER50	Driver Engine:		Driver Engine:		Driver Engine:
	IIS Server: 15.1.115.6 SQL Server:	DRIVER54	IIS Server: 15.1.115.7 SQL Server:	DRIVER58	IIS Server: 15.1.115.7 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 59853 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 64633 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 69413 -
61047	w_id Min Warehouse:	65827	w_id Min Warehouse:	70607	w_id Min Warehouse:
1		1		1	

99921	w_id Max Warehouse: Scale: Normal User Count: 11950 District id: 1 Scale Down: No Driver Engine:	1 99921	w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:	1 99921	w_id Min Warehouse: w_id Max Warehouse: Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:
DRIVER59	IIS Server: 15.1.115.7 SQL Server:	DRIVER63	IIS Server: 15.1.115.8 SQL Server:	DRIVER67	IIS Server: 15.1.115.8 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 70608 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 75382 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 80138 -
71802	w_id Min Warehouse:	76570	w_id Min Warehouse:	81326	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:
DRIVER60	IIS Server: 15.1.115.7 SQL Server:	DRIVER64	IIS Server: 15.1.115.8 SQL Server:	DRIVER68	IIS Server: 15.1.115.8 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 71803 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 76571 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 81327 -
72997	w_id Min Warehouse:	77759	w_id Min Warehouse:	82515	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:
DRIVER61	IIS Server: 15.1.115.7 SQL Server:	DRIVER65	IIS Server: 15.1.115.8 SQL Server:	DRIVER69	IIS Server: 15.1.115.8 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 72998 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 77760 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 82516 -
74192	w_id Min Warehouse:	78948	w_id Min Warehouse:	83704	w_id Min Warehouse:
1	w_id Max Warehouse:	1	w_id Max Warehouse:	1	w_id Max Warehouse:
99921	Scale: Normal User Count: 11950 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:	99921	Scale: Normal User Count: 11890 District id: 1 Scale Down: No Driver Engine:
DRIVER62	IIS Server: 15.1.115.8 SQL Server:	DRIVER66	IIS Server: 15.1.115.8 SQL Server:	DRIVER70	IIS Server: 15.1.115.8 SQL Server:
sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 74193 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 78949 -	sqloctanec	Database: tpcc User: sa Protocol: HTML w_id Range: 83705 -
75381		80137		84893	

1	w_id Min Warehouse:	89801	Protocol: HTML	93537 -	sqloctanec	SQL Server:
99921	w_id Max Warehouse:	1	w_id Min Warehouse:			Database: tpcc
	Scale: Normal	99921	w_id Max Warehouse:		94861	User: sa
	User Count: 11890		Scale: Normal		1	Protocol: HTML
	District id: 1		User Count: 12650		99921	w_id Range: 93597 -
	Scale Down: No		District id: 1			w_id Min Warehouse:
DRIVER71	Driver Engine:		Scale Down: No			w_id Max Warehouse:
	IIS Server: 15.1.115.8	DRIVER75	Driver Engine:			Scale: Normal
sqloctanec	SQL Server:	15.1.115.10	IIS Server:		DRIVER79	User Count: 12650
	Database: tpcc	sqloctanec	SQL Server:		15.1.115.10	District id: 1
	User: sa		Database: tpcc		sqloctanec	Scale Down: No
	Protocol: HTML		User: sa			Driver Engine:
	w_id Range: 84894 -		Protocol: HTML			IIS Server:
86082	w_id Min Warehouse:	91066	w_id Range: 89802 -			SQL Server:
1	w_id Max Warehouse:	1	w_id Min Warehouse:			Database: tpcc
99921	Scale: Normal	99921	w_id Max Warehouse:		96126	User: sa
	User Count: 11890		Scale: Normal		1	Protocol: HTML
	District id: 1		User Count: 12650		99921	w_id Range: 94862 -
	Scale Down: No		District id: 1			w_id Min Warehouse:
DRIVER72	Driver Engine:		Scale Down: No			w_id Max Warehouse:
	IIS Server: 15.1.115.8	DRIVER76	Driver Engine:			Scale: Normal
sqloctanec	SQL Server:	15.1.115.10	IIS Server:		DRIVER80	User Count: 12650
	Database: tpcc	sqloctanec	SQL Server:		15.1.115.10	District id: 1
	User: sa		Database: tpcc		sqloctanec	Scale Down: No
	Protocol: HTML		User: sa			Driver Engine:
	w_id Range: 86083 -		Protocol: HTML			IIS Server:
87271	w_id Min Warehouse:	92331	w_id Range: 91067 -			SQL Server:
1	w_id Max Warehouse:	1	w_id Min Warehouse:			Database: tpcc
99921	Scale: Normal	99921	w_id Max Warehouse:		97391	User: sa
	User Count: 11890		Scale: Normal		1	Protocol: HTML
	District id: 1		User Count: 12650		99921	w_id Range: 96127 -
	Scale Down: No		District id: 1			w_id Min Warehouse:
DRIVER73	Driver Engine:		Scale Down: No			w_id Max Warehouse:
	IIS Server:	DRIVER77	Driver Engine:			Scale: Normal
15.1.115.10	SQL Server:	15.1.115.10	IIS Server:		DRIVER81	User Count: 12650
sqloctanec	Database: tpcc	sqloctanec	SQL Server:		15.1.115.10	District id: 1
	User: sa		Database: tpcc		sqloctanec	Scale Down: No
	Protocol: HTML		User: sa			Driver Engine:
	w_id Range: 87272 -		Protocol: HTML			IIS Server:
88536	w_id Min Warehouse:	93596	w_id Range: 92332 -			SQL Server:
1	w_id Max Warehouse:	1	w_id Min Warehouse:			Database: tpcc
99921	Scale: Normal	99921	w_id Max Warehouse:		98656	User: sa
	User Count: 12650		Scale: Normal		1	Protocol: HTML
	District id: 1		User Count: 12650		99921	w_id Range: 97392 -
	Scale Down: No		District id: 1			w_id Min Warehouse:
DRIVER74	Driver Engine:		Scale Down: No			w_id Max Warehouse:
	IIS Server:	DRIVER78	Driver Engine:			Scale: Normal
15.1.115.10	SQL Server:	15.1.115.10	IIS Server:		DRIVER81	User Count: 12650
sqloctanec	Database: tpcc		SQL Server:		15.1.115.10	District id: 1
	User: sa		Database: tpcc		sqloctanec	Scale Down: No
			User: sa			Driver Engine:
			Protocol: HTML			IIS Server:
			w_id Range: 92332 -			SQL Server:
			w_id Min Warehouse:			Database: tpcc
			w_id Max Warehouse:			User: sa
			Scale: Normal			Protocol: HTML
			User Count: 12650			w_id Range: 97392 -
			District id: 1			w_id Min Warehouse:
			Scale Down: No			w_id Max Warehouse:
			Driver Engine:			Scale: Normal
			IIS Server:			User Count: 12650
			SQL Server:			District id: 1
			Database: tpcc			Scale Down: No
			User: sa			

DRIVER82	Driver Engine:	44.97	12.05	New Order	18.00	0.10
15.1.115.10	IIS Server:		5.00	Payment	43.05	
sqloctanec	SQL Server:		12.05		3.00	0.10
	Database: tpcc		5.00		0.10	
	User: sa		5.05	Delivery	4.01	
	Protocol: HTML		5.00		2.00	0.10
99921	w_id Range: 98657 -		5.05	Stock Level	4.01	
1	w_id Min Warehouse:		5.00		2.00	0.10
99921	w_id Max Warehouse:		10.05	Order Status	4.01	
	Scale: Normal		5.00		2.00	0.10
	User Count: 12650				0.10	
	District id: 1					
	Scale Down: No					

Number of Parameter Sets: 3

				~Default		
				Default Parameter Set		
				Txn		
Think	Key	RT	RT	Menu		Weight
Time	Time	Delay	Fence	Delay		
				New Order		
10.00		12.05		18.00		0.10
		5.00		0.10		
				Payment		10.00
		12.05		3.00		0.10
		5.00		0.10		
				Delivery		1.00
		5.05		2.00		0.10
		5.00		0.10		
				Stock Level		1.00
		5.05		2.00		0.10
		20.00		0.10		
				Order Status		1.00
		10.05		2.00		0.10
		5.00		0.10		

				Shutdown		
				Txn		
Think	Key	RT	RT	Menu		Weight
Time	Time	Delay	Fence	Delay		
				New Order		
10.00		100.00		2.00		0.10
		5.00		0.10		
				Payment		10.00
		100.00		2.00		0.10
		5.00		0.10		
				Delivery		1.00
		100.00		2.00		0.10
		5.00		0.10		
				Stock Level		1.00
		100.00		2.00		0.10
		5.00		0.10		
				Order Status		1.00
		100.00		2.00		0.10
		5.00		0.10		

				Audit		
				Txn		
Think	Key	RT	RT	Menu		Weight
Time	Time	Delay	Fence	Delay		

Appendix D 60 Day Space Requirements

TPC-C 60 Day Space Requirements							
Warehouses	100000	Rows	Data KB	Index KB	Extra 5% KB	TpmC	1,263,599.32
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB	
Warehouse	100000	10864	320	559		11743	
District	1000000	111304	472	5,589		117365	
Customer	3000000000	2181818368	136126344	115,897,236		2433841948	
History	3053632824	183628944	1310776		35,246,287	184939720	
NewOrder	905510520	23431488	74864	1,175,318		24681670	
Orders	3055570090	105563080	431552		40,106,570	105994632	
OrderLine	30555763307	2038163728	9043752		408,484,648	2047207480	
Item	100000	9416	120	477		10013	
Stock	10000000000	3200000000	6743744	160,337,187		3367080931	
Total		7,732,737,192	153,731,944	277,416,365	483,837,505	8,163,885,501	
		MB					
Dynamic Space	2,272,808	Sum of Data for Order, Orderline and History					
Static Space	5,699,736	Sum of Data+Index+5%-Dynamic Space					
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)					
Daily Growth	459,507	(Dynamic Space/(W*62.5))*tpmc					
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed					
60 Day Space MB	33,270,159						
60 Day Space GB	32,490.39	GB					
Log Size	1,850,000.00	MB					
KB Per New Order	6.00	KB					
8 hr log MB	3,553,441.55	MB					
8 hr log GB	3,470.16	GB					
		Disks	Disks	Formatted	Space		
Space Usage	GB Needed	Measured	Size	Size	Available		
180 Day Space DB	32,490.39	6	1550GB	1,550.000	9300.00		
		45	500GB	465.730	20957.85		
Total DB		51.00			30257.85		
8-hr log + mirror	6,940.3155	20	500GB	465.730	9314.60		
OS, Swap	3	2	72	68.330	136.66		
Total Storage Needed	39,433.71	GB			39,709.11		

3rd Party Pricing

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

May 16, 2011

Hewlett-Packard Company
Eric Deehr
One Microsoft Way
Redmond, WA 98052

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars (\$).

Part Number	Description	Unit Price	Quantity	Price
810-03134	SQL Server 2005 Enterprise Edition <i>Per Processor License Open Program - Level C Unit Price reflects a 6% discount from the retail unit price of \$24,999.</i>	\$23,432	4	\$93,728
P72-04217	Windows Server 2008 R2 Enterprise Edition <i>Server License with 25 CALs Open Program - Level C Unit Price reflects a 43% discount from the retail unit price of \$3,999.</i>	\$2,280	1	\$2,280
P73-04980	Windows Server 2008 R2 Standard Edition <i>Server License with 10 CALs Open Program - Level C Unit Price reflects a 31% discount from the retail unit price of \$1,029.</i>	\$711	8	\$5,688
N/A	Microsoft Problem Resolution Services <i>Professional Support (1 Incident).</i>	\$259	1	\$259

SQL Server 2005 Enterprise Edition, Windows Server 2008 R2 Enterprise Edition, and Windows Server 2008 R2 Standard Edition are currently orderable and available through Microsoft's normal distribution channels.

A list of Microsoft's resellers can be found in the Microsoft Product Information Center at <http://www.microsoft.com/products/info/render.aspx?view=22&type=how>
Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$259 call.
This quote is valid for the next 90 days.

Sales Quote



Violin Memory, Inc.
 27 Garcia Ave.
 Mountain View, CA 94043
 E. Casey Roche III
 Ph: 1-888-9VIOLIN (984-6546) x10
 Fax: 1 732-218-6077
[Email: casey@violin-memory.com](mailto:casey@violin-memory.com)
www.violin-memory.com

To: Eric Deehr
 1 Microsoft Way
 Redmond, WA 98052
 425-882-8080
eric.deehr@hp.com
 BL685c-TPC-C-BM

Date: May 12, 2011
 Quote #: ECR-051211HP8
 Expiration: Valid for
 Date: 30 days

Project BL685c-TPC-C-BM

Payment Terms: Net 30 days

Qty	Prod Code		List Price	Discount	Extended Price
Violin Memory Array					
4	V-3202	V-3200 Memory Array with 2.6TB Flash Redundant AC supplies - C19 connectors PClex8 Interface, Card and Cable	\$ 75,000.00	33%	\$ 201,000.00
2	V-3202A	V-3200 Chassis with PCIe 20 Gbit/s Interface 2.6TB Flash (42 x VIMM-64GSLC) Redundant AC supply PClex8 Interface Card and Cable	\$ 75,000.00	33%	\$ 100,500.00
Storage Interface					
2	VMG-8GFCx4	Violin Memory Gateway: 4 Port 4/8 Gbit/s Fiber Channel Interfaces	\$ 12,000.00	33%	\$ 16,080.00
Maintenance & Warranty					
3	VS-PM1Y	Violin Premium Maintenance - 1 year No specific software version requirements	4%	0%	\$ 56,880.00
Subtotal					\$ 374,460.00
Shipping & Handling					\$ -
Sales Tax					0.000% -
Total					\$ 374,460.00

Availability: Within 8 weeks after receipt of order.
 Shipping: FOB Origin, Shipping by 5 day ground within USA

Shipping Details	Eric Deehr
	1 Microsoft Way
	Redmond, WA 98052
	425-882-8080
	eric.deehr@hp.com
	BL685c-TPC-C-BM

Special Requirements: None

Thank You For Your Business!

2700 Garcia Ave, Mountain View, CA 94043 Ph: 1-888-9VIOLIN (984-6546) Email: sales@violin-memory.com www.violin-memory.com

System Availability

All hardware is currently available.

HP Direct: 800-203-6748

For price verification before order date: e-mail hp.pricing.desk@hp.com