



Hewlett-Packard Company

TPC Benchmark™ C
Full Disclosure Report
for
HP ProLiant DL370 G6
using
Microsoft SQL Server 2005 Enterprise x64 Edition SP2
and
Windows Server 2008 Enterprise x64 Edition SP2

First Edition
Submitted for Review
February 1, 2010

First Edition –February 2010

Hewlett-Packard Company (HP) 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. HP 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, HP 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 were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. HP 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 2010 Hewlett-Packard Company.

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 or on the title page of each item reproduced.

Printed in U.S.A., 2010

HP ProLiant DL370 G6 and ProLiant are registered trademarks of Hewlett-Packard Company.

Microsoft, Windows Server 2003, Windows Server 2008 x64 and SQL Server 2005 x64 are registered trademarks of Microsoft Corporation.

Xeon is a registered trademark of Intel.

TPC Benchmark is a trademark of the Transaction Processing Performance Council.

Other product names mentioned in this document may be trademarks and/or registered trademarks of their respective companies.

Table of Contents

TABLE OF CONTENTS	3
PREFACE	5
TPC BENCHMARK C OVERVIEW	5
ABSTRACT	6
OVERVIEW.....	6
TPC BENCHMARK C METRICS.....	6
STANDARD AND EXECUTIVE SUMMARY STATEMENTS	6
AUDITOR	6
GENERAL ITEMS	10
TEST SPONSOR.....	10
APPLICATION CODE AND DEFINITION STATEMENTS	10
PARAMETER SETTINGS	10
CONFIGURATION ITEMS	10
CLAUSE 1 RELATED ITEMS	12
TABLE DEFINITIONS	12
PHYSICAL ORGANIZATION OF DATABASE	12
<i>Benchmarked Configuration:</i>	12
PRICED CONFIGURATION VS. MEASURED CONFIGURATION:.....	15
INSERT AND DELETE OPERATIONS.....	15
PARTITIONING	16
REPLICATION, DUPLICATION OR ADDITIONS	16
CLAUSE 2 RELATED ITEMS	17
RANDOM NUMBER GENERATION.....	17
INPUT/OUTPUT SCREEN LAYOUT.....	17
PRICED TERMINAL FEATURE VERIFICATION.....	17
PRESENTATION MANAGER OR INTELLIGENT TERMINAL	17
TRANSACTION STATISTICS	18
QUEUEING MECHANISM	18
CLAUSE 3 RELATED ITEMS	19
TRANSACTION SYSTEM PROPERTIES (ACID)	19
ATOMICITY	19
<i>Completed Transactions</i>	19
<i>Aborted Transactions</i>	19
CONSISTENCY.....	19
ISOLATION	19
DURABILITY	20
<i>Durable Media Failure</i>	20
<i>Instantaneous Interruption and Loss of Memory</i>	21
CLAUSE 4 RELATED ITEMS	22
INITIAL CARDINALITY OF TABLES	22
DATABASE LAYOUT	22
TYPE OF DATABASE.....	22

DATABASE MAPPING	23
60 DAY SPACE.....	23
CLAUSE 5 RELATED ITEMS	24
THROUGHPUT	24
KEYING AND THINK TIMES.....	24
RESPONSE TIME FREQUENCY DISTRIBUTION CURVES AND OTHER GRAPHS	25
STEADY STATE DETERMINATION	30
WORK PERFORMED DURING STEADY STATE.....	30
MEASUREMENT PERIOD DURATION.....	30
REGULATION OF TRANSACTION MIX.....	31
TRANSACTION STATISTICS	31
CHECKPOINT COUNT AND LOCATION	32
CHECKPOINT DURATION.....	32
CLAUSE 6 RELATED ITEMS	33
RTE DESCRIPTIONS	33
EMULATED COMPONENTS	33
FUNCTIONAL DIAGRAMS	33
NETWORKS	33
OPERATOR INTERVENTION	33
CLAUSE 7 RELATED ITEMS	34
SYSTEM PRICING	34
AVAILABILITY, THROUGHPUT, AND PRICE PERFORMANCE.....	34
COUNTRY SPECIFIC PRICING	34
USAGE PRICING	34
CLAUSE 9 RELATED ITEMS	35
AUDITOR'S REPORT.....	35
AVAILABILITY OF THE FULL DISCLOSURE REPORT.....	35
APPENDIX A: SOURCE CODE	A-1 - A-111
APPENDIX B: DATABASE DESIGN	B-1 – B-52
APPENDIX C: TUNABLE PARAMETERS	C-1 - C-83
APPENDIX D: 60-DAY SPACE	D-1 - D-3
APPENDIX E: THIRD PARTY QUOTES	E-1 - E-4
APPENDIX F: PRICE VERIFICATION.....	F-1

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 5.10.1.

TPC Benchmark C Overview

The TPC describes this benchmark in Clause 0.1 of the specifications as follows:

TPC Benchmark™ C (TPC-C) is an OLTP workload. It is a mixture of read-only and update intensive transactions that simulate 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 on data access and update

The performance metric reported by TPC-C is a "business throughput" measuring 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.

Although these specifications express implementation in terms of a relational data model with conventional locking scheme, the database may be implemented using any commercially available database management system (DBMS), database server, file system, or other data repository that provides a functionally equivalent implementation. The terms "table", "row", and "column" are used in this document only as examples of logical data structures.

TPC-C uses terminology and metrics that are similar to other benchmarks, originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-C results are comparable to other benchmarks. The only benchmark results comparable to TPC-C are other TPC-C results conformant with the same revision.

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 any other environment 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 benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the HP ProLiant DL370 G6. The operating system used for the benchmark was Windows Server 2008, Enterprise x64 Edition SP2. The DBMS used was Microsoft SQL Server 2005 Enterprise x64 Edition SP2.

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:

661,475 tpmC

USD \$1.16 per tpmC

The availability date is February, 2010.

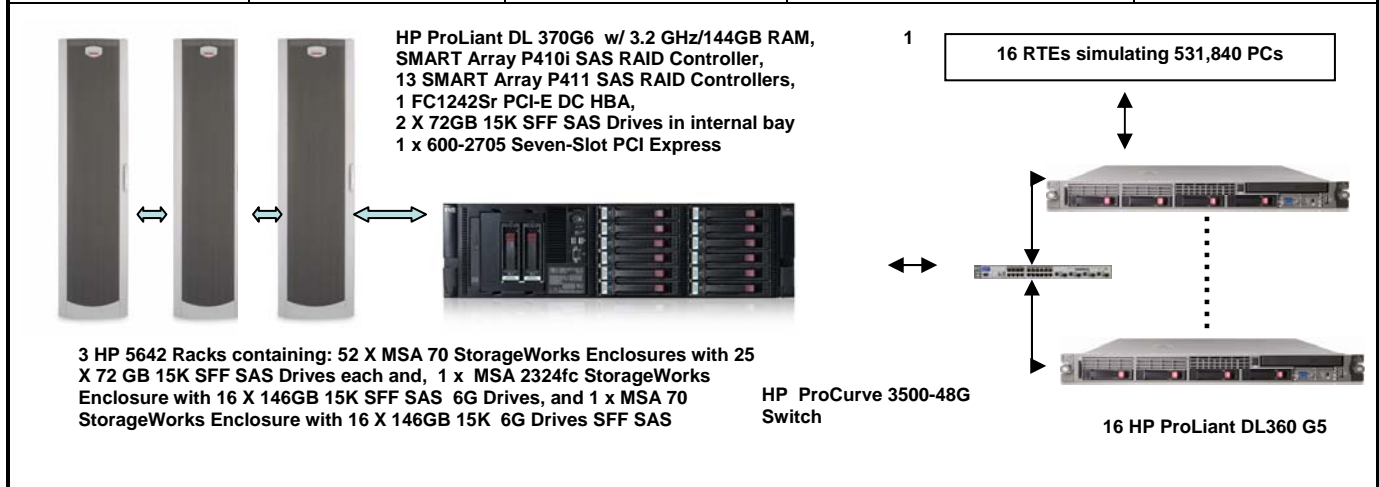
Standard and Executive Summary Statements

The following pages contain executive summary of results for this benchmark.

Auditor

The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

Hewlett-Packard Company	HP ProLiant DL370G6 3.2 GHz 8MB L2		TPC-C Rev. 5.10.1	
	C/S with 16 HP ProLiant DL360 G5		Report Date: Feb 1, 2010	
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
USD \$767,082	661,475	USD \$1.16	Feb 1, 2010	
Database Server Processors /Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
2/8/16 Intel Xeon 3.2 GHz 8MB L2 cache	Microsoft SQL Server 2005 Enterprise x64 Edition SP2	Windows Server 2008 SP2 Enterprise x64 Edition	Microsoft Visual C++ Microsoft COM+	531,840



	Server		Each Client	
System Components	Quantity	Description	Quantity	Description
Processors/Cores/Threads	2/8/16	Intel Xeon 3.2GHz 8MB L2 cache	1/4/4	2.83 GHz Intel Xeon w/ 12MB L2 cache
Memory	144GB	(18 x 8GB) DDR3	1GB	1024 MB
Disk Controllers	1 13 1	Smart P410i Controller Smart P411 Controller FC1242SR PCI-E HBA	1	Integrated Smart Array 400i Controller
Disk Drives	32 1300 2	146GB 15K SFF SAS 6G 72 GB 15K SFF SAS 72 GB 15K SFF SAS	2	72 GB 15K SFF SAS
Total Storage		92,389.60 GB		72 GB

Hewlett-Packard Company	HP ProLiant DL370G6			TPC-C Rev. 5.10.1		
				Report Date	1-Feb-10	
Description	Part Number	Brand	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware						
HP ProLiant DL370 G6 SFF CTO Chassis	483874-B21	1	1,662	1	1,662	
W5580 3.20 GHz, 8MB L3 Cache, 130W	495928-L21	1	2,299	1	2,299	
W5580 3.20 GHz, 8MB L3 Cache, 130W	495928-B21	1	2,299	1	2,299	
HP 8GB 2Rx4 PC3-8500R-7 Kit	516423-B21	1	990	18	17,820	
HP 750W CS HE Gold Power Supply	512327-B21	1	299	2	598	
HP P411/512 BBWC Smart Array Controller	462832-B21	1	649	13	8,437	
HP StorageWorks FC1242 Dual Channel 4Gb PCI-e HBA	AE312A	1	1,780	1	1,780	
2 m LC-LC Multi-Mode Fibre Channel Cable	221692-B21	1	75	2	150	
2 m LC-LC Multi-Mode Fibre Channel Cable	221692-B21	1	75	2		150
HP NC364T PCI-E Quad Port Gigabit Server Adapter	435508-B21	1	529	1	529	
HP StorageWorks 2324fc G2 Dual Controller Modular Smart Array (SFF)	AJ797A	1	8,900	1	8,900	
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	
HP 5642 Pallet Unassembled Rack	358254-B21	1	865	3	2,595	
HP R1.5 kVA 1U NA UPS	AF419A	1	739	1	739	
HP 72GB 15k 2.5 dual Port HP SAS Drive	418371-B21	1S	349	1300	453,700	
HP 72GB 15k 2.5 dual Port HP SAS Drive (10% Spares)	418371-B21	1S	349	130		45,370
HP 146GB 6G SAS 15K SFF DP ENT HDD	512547-B21	1	579	32	18,528	
HP 146GB 6G SAS 15K SFF DP ENT HDD (10% Spares)	512547-B21	1	579	4		2,316
HP 72GB 15k 2.5 dual Port HP SAS Drive	418371-B21	1S	349	2	698	
HP StorageWorks MSA-70 Storage	418800-B21	1	3,199	53	169,547	
HP StorageWorks MSA-70 Storage (10% Spares)	418408-B21	1	3,199	6		19,194
600-2705 System: Low Latency Seven-Slot PCI Express	600-2705-1-SH	4	1,524	1	1,524	
600-2705 System: Low Latency Seven-Slot PCI Express 10% spares	600-2705-1-SH	4	1,524	2		3,048
HP 3y 4h 24x7 ProLiant DL370 HW Support ,ProLiant Server DL370	UM753E	1	931	1		931
Subtotal					692,003	71,009
Server Software						
Microsoft SQL Server 2005 Enterprise X64 Edition(per processor)	810-03134	2	23,432	2	46,864	Incl Below
Visual Studio Standard 2005	127-00012	2	250	1	250	Incl Below
Microsoft Windows Server 2008 Enterprise Edition (x64)	P72-03168	2	2,301	1	2,301	Incl Below
Microsoft Problem Resolution Services		2	245	1		245
Subtotal					49,415	245
Client Hardware						
HP DL360R05 E5440 2G US Svr	457923-001	1	2,899	16	46,384	
Dual Integrated Gigabit NIC, HP Smart Array P400i/256MB Controller						
HP 72GB 15k 2.5 dual Port HP SAS Drive	418371-B21	1S	349	32	11,168	
HP LE1851w 18.5-Inch wide Monitor	NK033AA#ABA	1	159	16	2,544	
HP PS/2 Keyboard And Mouse Bundle	RC464AA#ABA	1	39	16	624	
HP CP 3Y 4H 24x7 HW Entry300 4-Hour 24 Hour x 7 Day Coverage 3 Years	U4497E	1	698	16		11,168
Subtotal					60,720	11,168
Client Software						
Windows Server 2003 R2 Standard Edition	P73-01972	2	719	16	11,504	Incl. Above
Subtotal					11,504	0
User Connectivity						
HP ProCurve Switch 3500-48G	J9472A#ABA	1	4,099	1	4,099	
3-year, 4-hour onsite, 24x7 coverage for hardware	H2893E	1	1,307	1		1,307
CAT 6 7 Foot Gray Patch Cable	CB242-7G	3	2	34	54	
CAT 6 7 Foot Gray Patch Cable	CB242-7G	3	2	4		6
Subtotal					4,153	1,313
Large Purchase and Net 30 discount (See Note 1)	16.0%	1			(\$121,092)	(\$13,357)
Total					\$696,704	\$70,378
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 pricing specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org . Thank you.					Three-Year Cost of Ownership: USD \$767,082	
Pricing: 1=HP Direct 800-203-6748 2=Microsoft 3= deepsurplus.com 4= Cyclone Microsystems					tpmC Rating: 661,475	
Note 1 = Discount based on HP Direct guidance applies to all lines where pricing = 1					\$ / tpmC: USD \$1.16	
Note 2 = (S) One or more component of the measured configuration have been substituted in the priced configuration. See FDR for details.						
Note 3 = The benchmark results were audited by Lorna Livingtree of Performance Metrics						

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput

661,475 tpmC

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.52	1.11	8.39
Payment	0.48	1.07	4.65
Order-Status	0.51	1.10	8.41
Delivery (interactive portion)	0.11	0.11	2.70
Delivery (deferred portion)	0.14	0.22	4.63
Stock-Level	0.56	1.19	4.29
Menu	0.11	0.11	3.07

Transaction Mix, in percent of total transaction

New-Order	44.93%
Payment	43.04%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.01%

Emulation Delay (in seconds)

	Resp.Time	Menu
New-Order	0.10	0.10
Payment	0.10	0.10
Order-Status	0.10	0.10
Delivery (interactive)	0.10	0.10
Stock-Level	0.10	0.10

Keying/Think Times (in seconds)

	Min.	Average	Max.
New-Order	18.02/0.00	18.03/12.06	19.18/120.53
Payment	3.02/0.00	3.03/12.06	4.19/121.48
Order-Status	2.02/0.00	2.03/10.06	3.15/100.53
Delivery (interactive)	2.02/0.00	2.03/5.07	3.18/50.53
Stock-Level	2.02/0.00	2.03/5.06	3.16/50.53

Test Duration

Ramp-up time	44 minutes
Measurement interval	120 minutes
Transactions (all types) completed during measurement interval	183,754,059
Ramp down time	38 minutes

Checkpointing

Number of checkpoints	4
Checkpoint interval	30 minutes

General Items

Test Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by Hewlett-Packard Company. The benchmark was developed and engineered by Hewlett-Packard Company. Testing took place at HP benchmarking laboratories in Houston, Texas.

Application Code and Definition Statements

The application program (as defined in clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input output functions.

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

- *Database options*
- *Recover/commit options*
- *Consistency locking options*
- *Operating system and application configuration parameters*

This requirement can be satisfied by providing a full list of all parameters.

Appendix C contains the tunable parameters to for the database, the operating system, and the transaction monitor.

Configuration Items

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

The configuration diagram for both the tested and priced systems are included on the following page.

Figure 1. Benchmarked Configuration

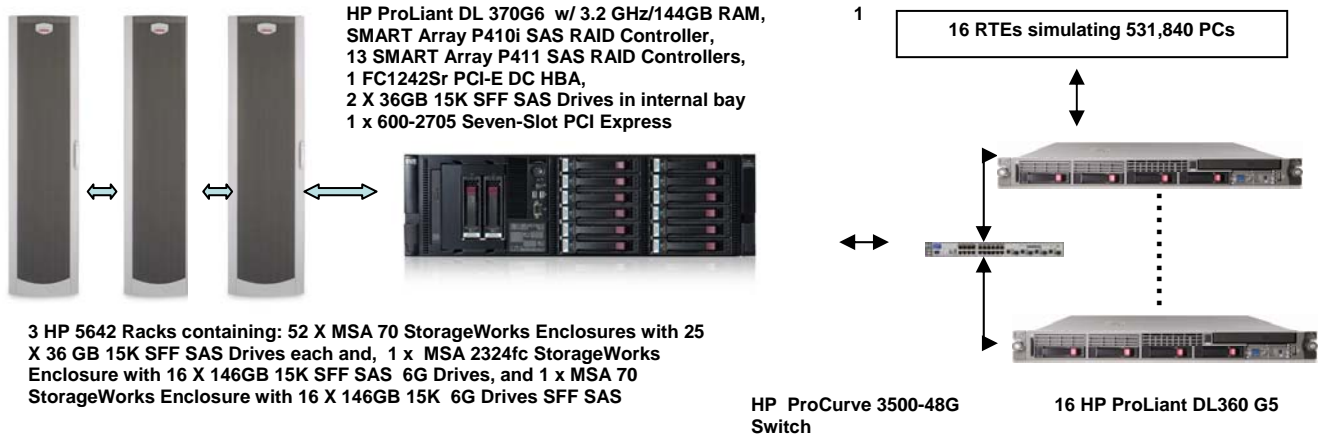
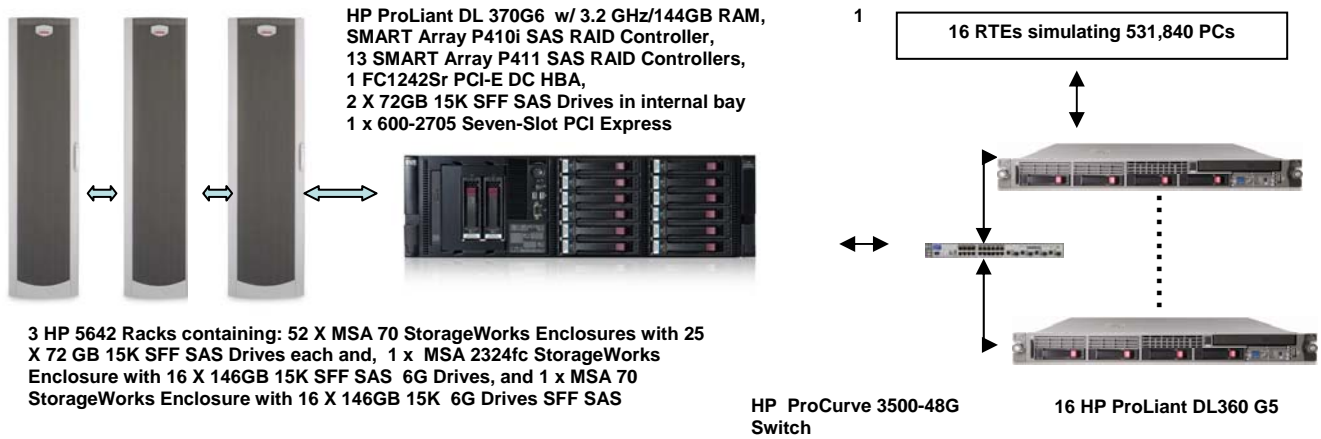


Figure 2. Priced Configuration



Clause 1 Related Items

Table Definitions

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.

Physical Organization of Database

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

The tested configuration consisted of 1300 drives at 36GB for database data, two 36GB drives for the operating system, and 32 drives at 146GB 6G for database log. There were 1300 X 36GB drives for database data on thirteen SMART P411 controllers, 32 X 146 GB drives on one FC1242 PCI-E HBA connecting 1 MSA 2324fc and 1 MSA70 for database log, and 2 X 36GB drives on the SMART P410i controller for the operating system.

Benchmarked Configuration:

SMART-P400 Controller, Slot 0, Array A

LOGICAL DRIVE C: Total Capacity = 33.91 GB RAID 0+1
Microsoft Windows Server 2008 Enterprise X64 Edition SP2

SMART-P411 Controller, Slot 1, Array A

LOGICAL DRIVE C:\stk\stk1: Total Capacity = 166.01 GB RAID 0
Stk_fg
LOGICAL DRIVE C:\cust\cust1: Total Capacity = 104.49 GB RAID 0
Cust_fg
LOGICAL DRIVE C:\ol\ol1: Total Capacity = 118.16GB RAID 0
ol_fg
LOGICAL DRIVE C:\misc\misc1: Total Capacity = 29.29 GB RAID 0
Misc_fg
LOGICAL DRIVE N: Total Capacity = 1485.37 GB RAID 0+1
Tpcback11

SMART-P411 Controller, Slot 2, Array A

LOGICAL DRIVE C:\stk\stk2: Total Capacity = 166.01 GB RAID 0
Stk_fg
LOGICAL DRIVE C:\cust\cust2: Total Capacity = 104.49 GB RAID 0
Cust_fg
LOGICAL DRIVE C:\ol\ol2: Total Capacity = 118.16GB RAID 0
ol_fg
LOGICAL DRIVE C:\misc\misc2: Total Capacity = 29.29 GB RAID 0
Misc_fg
LOGICAL DRIVE O: Total Capacity = 1485.37 GB RAID 0+1
Tpcback12

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk3:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust3:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol3:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc3:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE P:</u> Tpcback13	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk4:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust4:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol4:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc4:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE Q:</u> Tpcback6	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk5:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust5:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol5:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc5:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE R:</u> Tpcback7	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk6:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust6:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol6:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc6:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE S:</u> Tpcback8	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk7:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust7:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol7:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc7:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE T:</u> Tpcback9	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 3, PCI-E Expansion Box Array A

<u>LOGICAL DRIVE C:\stk\stk8:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust8:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol8:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc8:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE U:</u> Tpcback10	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 4, Array A

<u>LOGICAL DRIVE C:\stk\stk9:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust9:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol9:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc9:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE V:</u> Tpcback1	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 5, Array A

<u>LOGICAL DRIVE C:\stk\stk10:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust10:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol10:</u> ol_fg	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\misc\misc10:</u> Misc_fg	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE W:</u> Tpcback2	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>

SMART-P411Controller, Slot 6, Array A

<u>LOGICAL DRIVE C:\stk\stk11:</u> Stk_fg	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\cust\cust11:</u> Cust_fg	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
<u>LOGICAL DRIVE C:\ol\ol11:</u>	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>

ol_fg LOGICAL DRIVE C:\misc\misc11:	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
Misc_fg LOGICAL DRIVE X:	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>
Tpcback3		

SMART-P411Controller, Slot 7, Array A

LOGICAL DRIVE C:\stk\stk12:	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
Stk_fg LOGICAL DRIVE C:\cust\cust12:	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
Cust_fg LOGICAL DRIVE C:\ol\ol12:	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
ol_fg LOGICAL DRIVE C:\misc\misc12:	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
Misc_fg LOGICAL DRIVE Y:	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>
Tpcback4		

SMART-P411Controller, Slot 8, Array A

LOGICAL DRIVE C:\stk\stk13:	<u>Total Capacity = 166.01 GB</u>	<u>RAID 0</u>
Stk_fg LOGICAL DRIVE C:\cust\cust13:	<u>Total Capacity = 104.49 GB</u>	<u>RAID 0</u>
Cust_fg LOGICAL DRIVE C:\ol\ol13:	<u>Total Capacity = 118.16GB</u>	<u>RAID 0</u>
ol_fg LOGICAL DRIVE C:\misc\misc13:	<u>Total Capacity = 29.29 GB</u>	<u>RAID 0</u>
Misc_fg LOGICAL DRIVE Z:	<u>Total Capacity = 1485.37 GB</u>	<u>RAID 0+1</u>
Tpcback5		

SMART- FC1242SR PCI-E HBA Controller / MSA2324, Slot 9,

VD1:	<u>Total Capacity = 1173.2 GB</u>	<u>RAID 10</u>
VD2:	<u>Total Capacity = 1173.2 GB</u>	<u>RAID 10</u>

Note: The two mirrored vdisks were configured as one dynamic spanned volume and one simple volume in the operating system.

LOGICAL DRIVE E: MSSQL_tpc_log_1	<u>Total Capacity = 2034.54 GB</u>	<u>RAID 10</u>
-------------------------------------	------------------------------------	----------------

LOGICAL DRIVE E: MSSQL_tpc_log_2	<u>Total Capacity = 150.34 GB</u>	<u>RAID 10</u>
-------------------------------------	-----------------------------------	----------------

Priced Configuration vs. Measured Configuration:

The benchmarked configuration was run using all 36GB 15k drives for data and operating systems. The priced configuration substituted all data and operating system 36GB 15k drives for 72GB 15k drives.

Insert and Delete Operations

It must be ascertained that insert and/or 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 minimum key value for these new rows.

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

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.

No partitioning was used in this benchmark.

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 in this benchmark.

Clause 2 Related Items

Random Number Generation

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

In the Benchcraft RTE from Microsoft, each driver engine uses an independent random number sequence. All of the users within a given driver draw from the same sequence.

The Benchcraft RTE computes random integers as described in "Random Numbers Generators: Good Ones Are Hard to Find." Communications of the ACM - October 1988 Volume 31 Number 10.

The seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched, and randomly sampled by the auditor for patterns that would indicate the random number generator had affected any kind of a discernible pattern; none was found.

Input/Output Screen Layout

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

All screen layouts followed the specifications exactly.

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 attributes were verified by the auditor. The auditor manually exercised each specification on a representative HP ProLiant web server.

Presentation Manager or Intelligent Terminal

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

Application code running on the client machines implemented the TPC-C user interface. No presentation manager software or intelligent terminal features were used. The source code for the forms applications is listed in Appendix A.

Transaction Statistics

Table 2.1 lists the numerical quantities that Clauses 8.1.3.5 to 8.1.3.11 require.

Table 2.1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse payments	85.00%
	Remote warehouse payments	15.00%
	Accessed by last name	60.00%
Order Status	Accessed by last name	60.07%
Transaction Mix	New Order	44.93%
	Payment	43.04%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.01%

Queuing Mechanism

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

Microsoft COM+ on each client machine served as the queuing mechanism to the database. Each delivery request was submitted to Microsoft COM+ asynchronously with control being returned to the client process immediately and the deferred delivery part completing asynchronously.

The source code is listed in Appendix A.

Clause 3 Related Items

Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

All ACID property tests were successful. The executions are described below.

Atomicity

The system under test must guarantee that the database transactions are atomic; the system will either perform all individual operations on the data or will assure that no partially completed operations leave any effects on the data.

Completed Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

Aborted Transactions

A row was selected in a script from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

Consistency

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

Consistency conditions one through four were tested using a script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A run was executed under full load lasting over two hours and included a checkpoint.

The script was executed again. The result of the same queries verified that the database remained consistent after the run.

Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

Isolation tests one through nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

In addition, the phantom tests and the stock level tests were executed and verified.

For Isolation test seven, case A was followed.

Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transaction and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

Durable Media Failure

Loss of Data and Log

To demonstrate recovery from a permanent failure of durable medium containing DBMS logs and TPC-C tables, the following steps were executed. This test was executed on a fully scaled database of 54000 warehouses of which 5400 were used under a load of 54000 users.

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTEs were started with 54000 users.
- The test was allowed to run for a minimum of 10 minutes.
- One disk was removed from the MSA 2324fc containing the log disks.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the user's status on the RTE.
- One of the data disks was removed from one MSA 70 data drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down, and a database transaction log dump was taken.
- Microsoft SQL Server was shutdown, and the system rebooted after replacing the pulled drives with new drives.
- After the RAID recovery process finished Microsoft SQL Server was started.
- The database was restored from backup and the transaction log dump was applied.
- Consistency condition #3 was executed and verified.
- Step 2 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in steps 12 and 13 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database of 53184 warehouses under a full load of 531840 users. The following steps were executed:

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 508320 users.
- The test was allowed to run for a minimum of 6 minutes.
- Pulling the power cords from the SUT induced system crash and loss of memory. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was paused then stopped.
- Power was restored and the system restarted.
- Microsoft SQL Server was restarted and performed an automatic recovery.
- Consistency condition #3 was executed and verified.
- Step 1 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 9 and 10 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Clause 4 Related Items

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 4.1 Number of Rows for Server

Table	Cardinality as built
Warehouse	54000
District	540,000
Customer	1,620,000,000
History	1,620,000,000
Orders	1,620,000,000
New Order	486,000,000
Order Line	16,199,947,865
Stock	5,600,000,000
Item	100,000
Unused Warehouses	816

Database Layout

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

The benchmarked configuration used 1300 SAS drives at 36GB for database data, two 36GB SAS drives for the operating system, and 32 SAS drives at 146GB 6G for database log. Thirteen SMART P411 controllers connected to 2 MSA70 drive boxes per port for each of two ports. Each MSA70 contained (25) 36GB SAS drives. Each controller was configured in an array. Each array had four RAID 0 logical drives for data, and a RAID 0+1 logical drive for database backup files. The SMART P410i controller was connected to the internal drive cage which contained 2 X 36GB SAS drives configured as a RAID 0+1 logical drive. The Array Accelerators on the data controllers were configured as 100% write cache and were enabled for all logical drives. The FC1242 HBA connected an MSA2324fc with dual controllers and one MSA 70 for transaction log. The MSA 2324 also had the controller caches enabled at 100% write in fault tolerant mode for log. All RAID volumes used hardware RAID.

Section 1.2 of this report details the distribution of database tables across all disks. The code that creates the file groups and tables is included in Appendix B.

Type of Database

A statement must be provided that describes:

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

- *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 transaction. 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 is a relational DBMS.

The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code.

Database Mapping

The mapping of database partitions/replications must be explicitly described.

The database was not replicated.

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 (Order, Order-Line, and History) must be disclosed.

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

- The free space on the log file was queried using *dbcc sqlperf(logspace)*.
- Transactions were run against the database with a full load of users.
- The free space was again queried using *dbcc sqlperf(logspace)*.
- The space used was calculated as the difference between the first and second query.
- The number of NEW-ORDERS was verified from the difference in the sum(d_next_o_id) taken from before and after the run.
- The space used was divided by the number of NEW-ORDERS giving a space used per NEW-ORDER transaction.
- The space used per transaction was multiplied by the measured tpmC rate times 480 minutes.

The same methodology was used to compute growth requirements for dynamic tables Order, Order-Line and History.

Details of both the 8-hour transaction log space requirements and the 60-day space requirements are shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 661,475 tpmC

Price per tpmC USD \$1.16

Response Times

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

Table 5.2: Response Times

Type	Average	90 th %	Maximum
New-Order	0.52	1.11	8.39
Payment	0.48	1.07	4.65
Order-Status	0.51	1.10	8.41
Interactive Delivery	0.11	0.11	2.70
Deferred Delivery	0.14	0.22	4.63
Stock-Level	0.56	1.19	4.29
Menu	0.11	0.11	3.07

Keying and Think Times

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

Table 5.3: Keying Times

Type	Minimum	Average	Maximum
New-Order	18.02	18.03	19.18
Payment	3.02	3.03	4.19
Order-Status	2.02	2.03	3.15
Interactive Delivery	2.02	2.03	3.18
Stock-Level	2.02	2.03	3.16

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.06	120.53
Payment	0.00	12.06	121.48
Order-Status	0.00	10.06	100.53
Interactive Delivery	0.00	5.07	50.53
Stock-Level	0.00	5.06	50.53

Response Time Frequency Distribution Curves and Other Graphs

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.

Figure 3. New Order Response Time Distribution

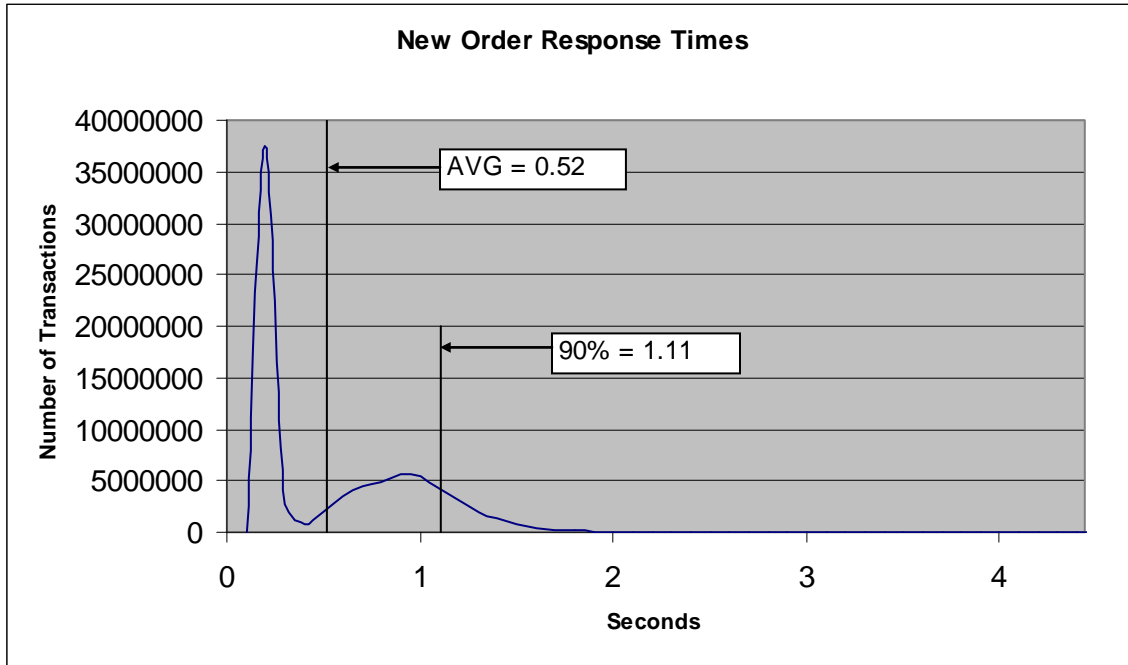


Figure 4. Payment Response Time Distribution

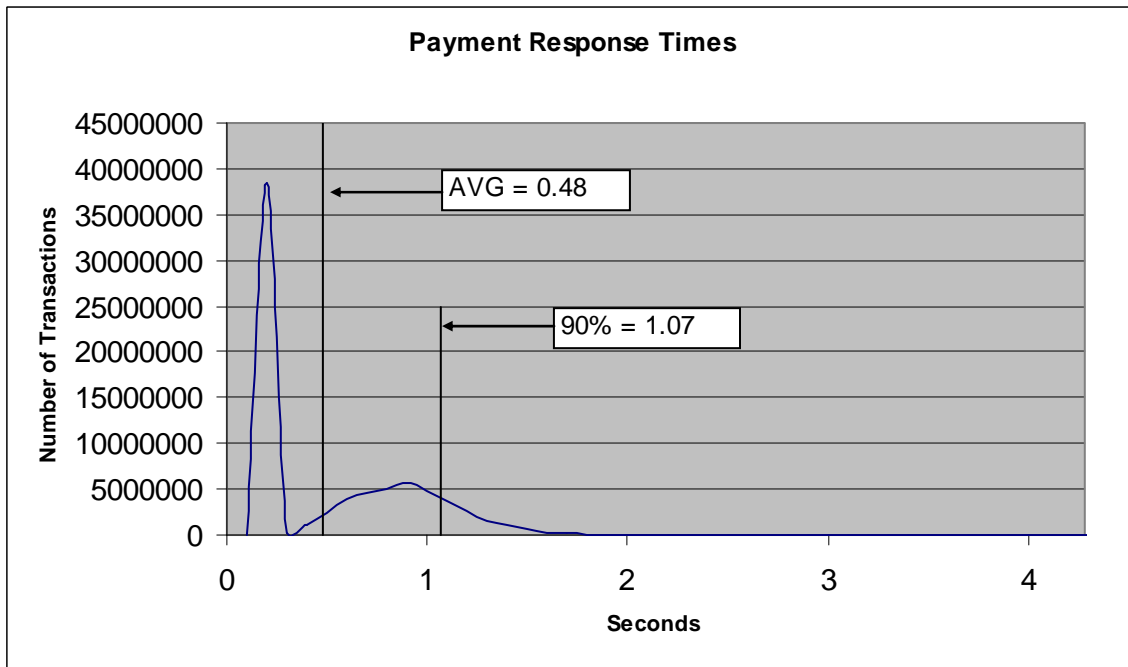


Figure 5. Order Status Response Time Distribution

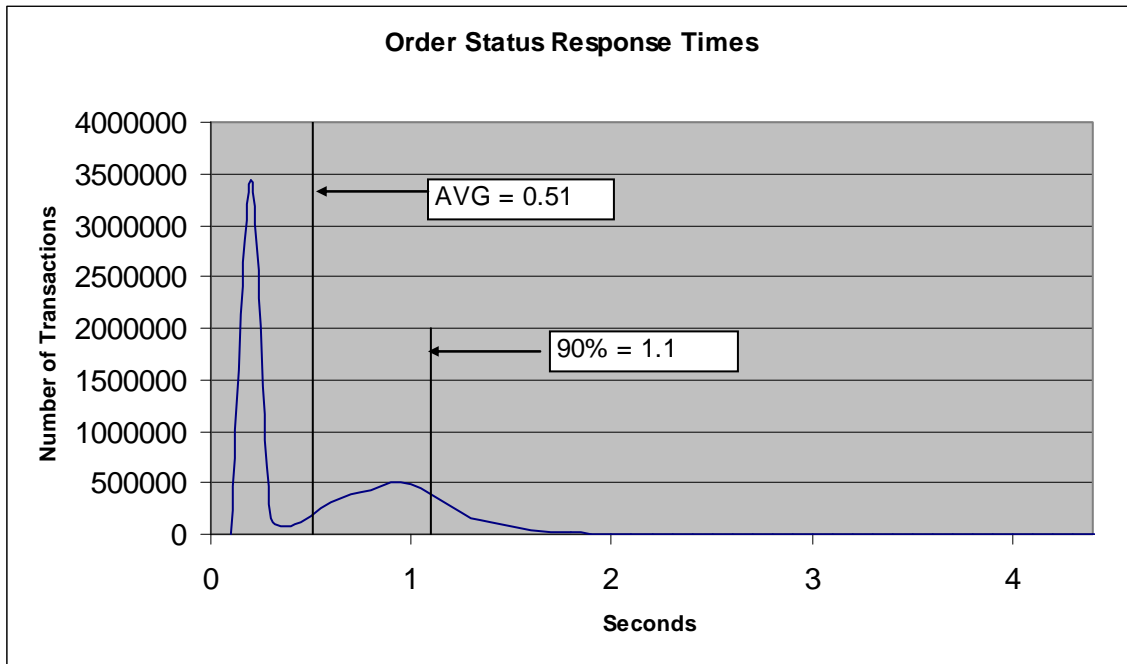


Figure 6. Delivery Response Time Distribution

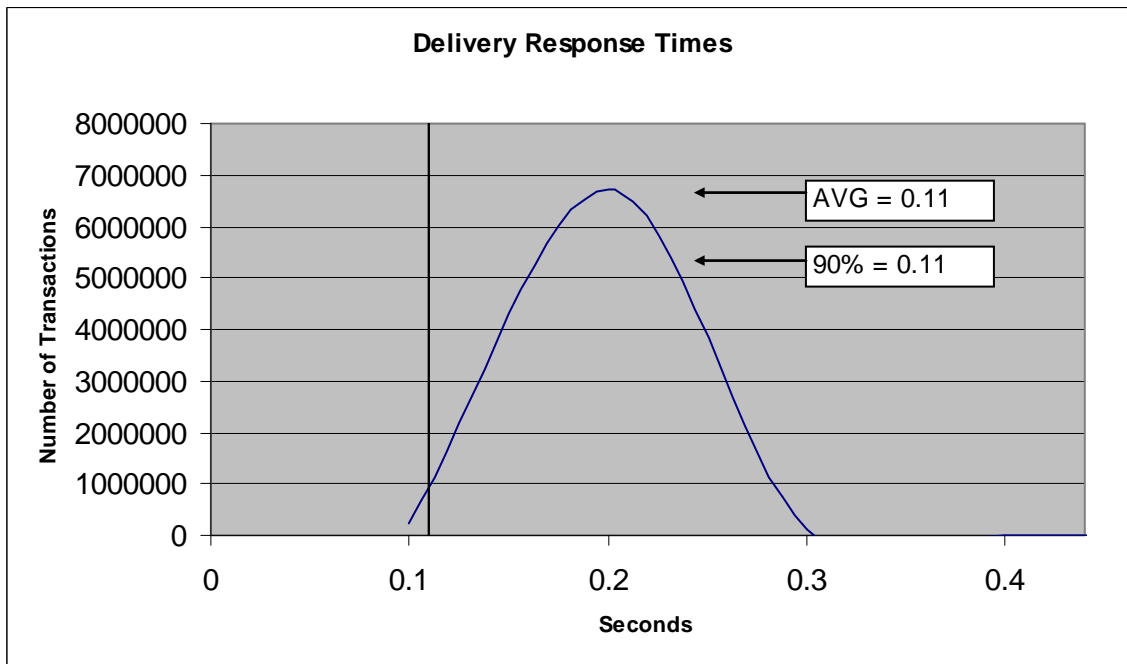


Figure 7. Stock Level Response Time Distribution

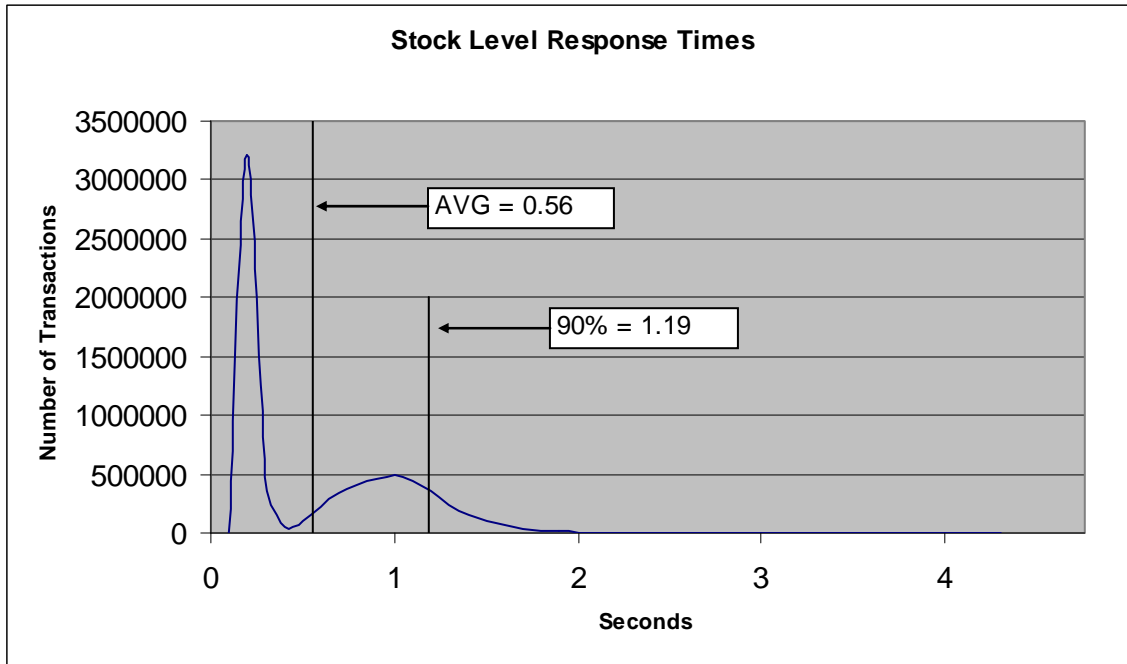


Figure 8. Response Time vs. Throughput

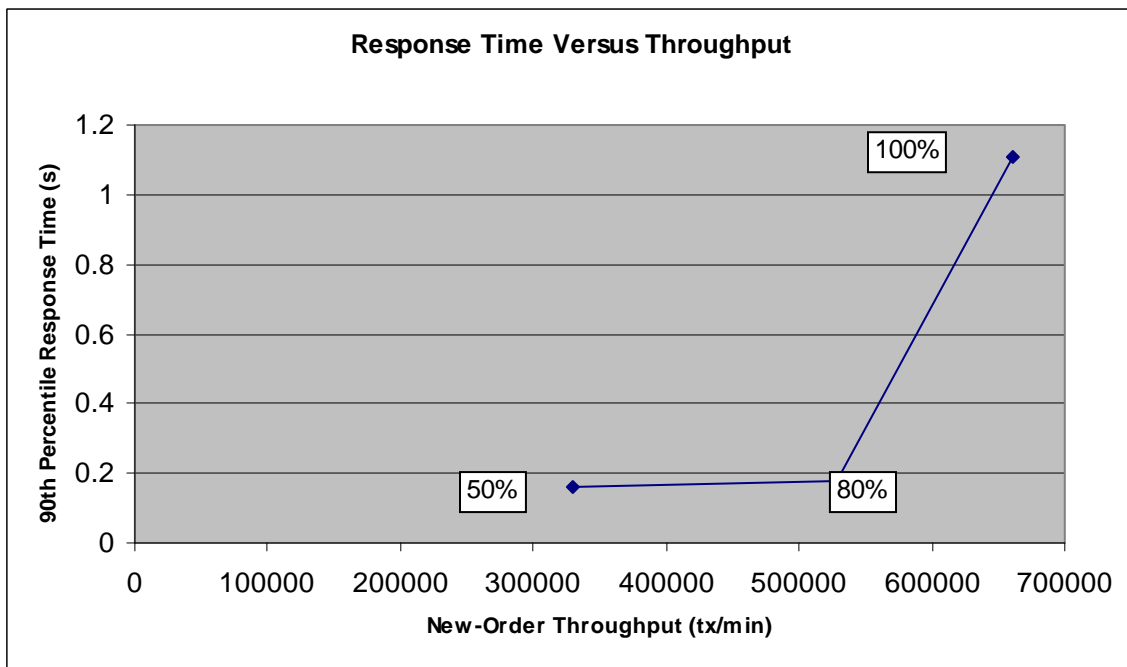


Figure 9. New Order Think Time Distribution

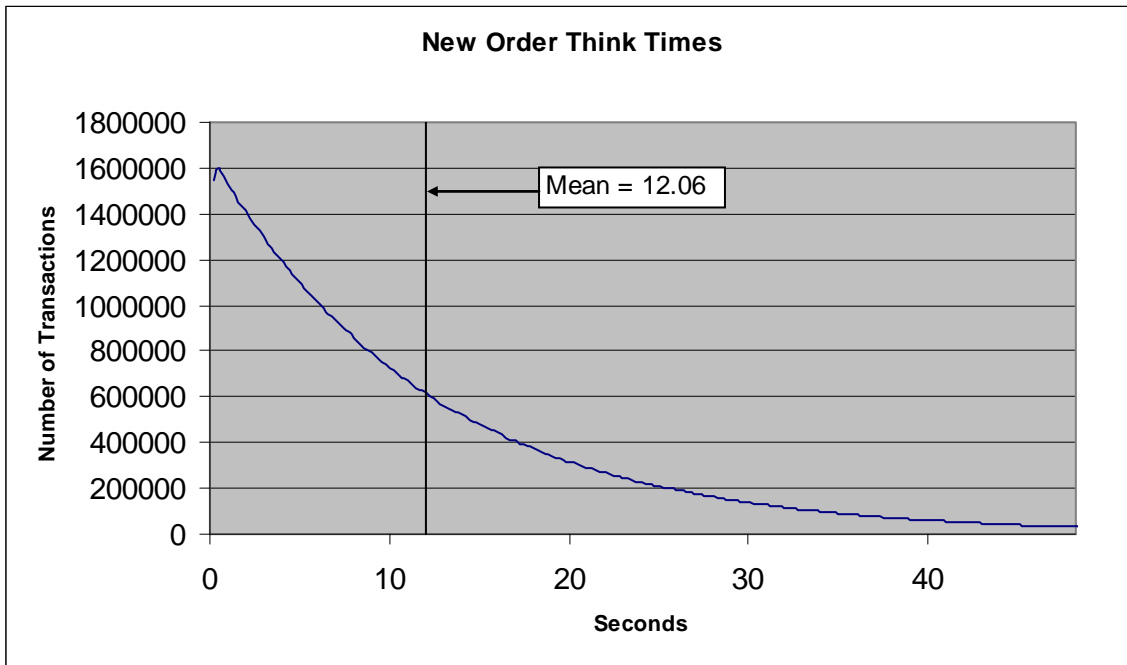
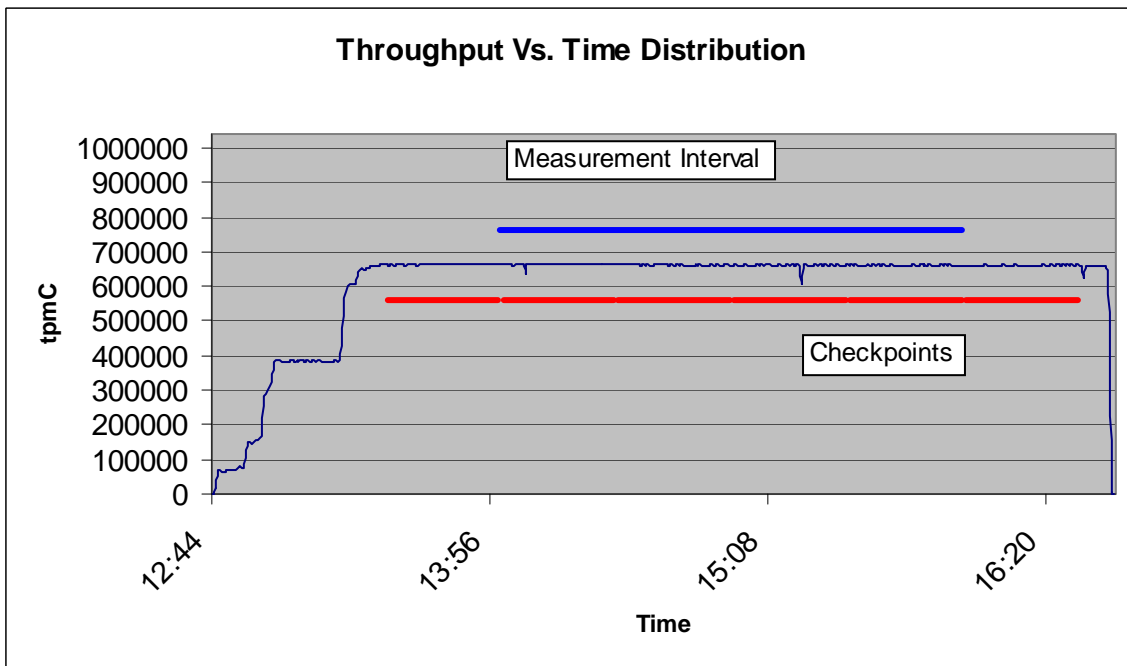


Figure 10. Throughput vs. Time Distribution



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.

Steady state was determined using real time monitor utilities from the RTE. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 10.

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 RTE generated the required input data to choose a transaction from the menu. This data was time stamped. The input screen for the requested transaction was returned and time stamped. The difference between these two timestamps was the menu response time. The RTE writes to the log file once per transaction on selective fields such as order id. There is one log file per driver engine.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was time stamped. The return of the screen with the required response data was time stamped. The difference between these two timestamps was the response time for that transaction.

The RTE then waited the required think time interval before repeating the process starting at selecting a transaction from the menu.

The RTE transmissions were sent to application processes running on the client machines through Ethernet LANs. These client application processes handled all screen I/O as well as all requests to the database on the server. The applications communicated with the database server over gigabit Ethernet LANs using ODBC and RPC calls.

To perform checkpoints at specific intervals, the SQL Server *recovery interval* was set to 32767 and a script was written to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to 30 minutes. The measurement interval was 120 minutes. The checkpoint script was started manually after the RTE had all users logged in and the database had achieved steady state.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. The positioning of the measurement interval is depicted on the graph in Figure 9.

Measurement Period Duration

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

The reported measured interval was exactly 120 minutes long.

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 RTE was given a weighted random distribution, which was not adjusted during the run.

Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. 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 per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections 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 5.5: Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse payments	85.00%
	Remote warehouse payments	15.00%
	Accessed by last name	60.00%
Delivery	Skipped transactions (interactive)	0
	Skipped transactions (deferred)	0
Order Status	Accessed by last name	60.07%
Transaction Mix	New Order	44.93%
	Payment	43.04%
	Order status	4.01%
	Delivery	4.01%
	Stock level	4.01%

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 initial checkpoint was started 44 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 30 minutes. Each checkpoint in the measurement interval lasted 28 minutes and 45 seconds. The measurement interval contains four checkpoints.

Checkpoint Duration

The start time and duration in seconds of at least the four longest checkpoints during the Measurement Interval must be disclosed.

Checkpoint Start Time	Duration
14:00:17 PM	28 minutes, 45 seconds
14:30:14 PM	28 minutes, 45 seconds
15:00:11 PM	28 minutes, 45 seconds
15:30:08 PM	28 minutes, 45 seconds

Clause 6 Related Items

RTE Descriptions

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used.

The RTE used was Microsoft Benchcraft RTE. Benchcraft is a proprietary tool provided by Microsoft and is not commercially available. The RTE's input is listed in Appendix A.

Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

The driver system consisted of 16 HP ProLiant servers. These driver machines emulated the users' web browsers.

Functional Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all hardware and software functionality being performed on the Driver System and its interface to the SUT must be disclosed.

The driver system performed the data generation and input functions of the priced display device. It also captured the input and output data and timestamps for post-processing of the reported metrics. No other functionality was included on the driver system.

Section 1.4 of this report contains detailed diagrams of both the benchmark configuration and the priced configuration.

Networks

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

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

In the tested configuration, 16 driver (RTE) machines were connected through a gigabit Ethernet switch to the client machines at 1Gbps, thus providing the path from the RTEs to the clients. The server (SUT) was connected to the clients through a gigabit Ethernet switch on a separate LAN.

The priced configuration was connected in the same manner as the tested configuration.

Operator Intervention

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

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

Clause 7 Related Items

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, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The details of the hardware and software are reported in the front of this report as part of the executive summary. All third party quotations are included at the end of this report as Appendix E.

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 included 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 (price/tpmC), and the availability date must be included.

- **Maximum Qualified Throughput** **661,475tpmC**
- **Price per tpmC** **USD \$1.16 per tpmC**
- **Availability** **February 1, 2010**

Country Specific Pricing

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

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

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:

- 16 Microsoft Windows Server 2003 Standard Edition R2
- 1 Microsoft Windows Server 2008 Enterprise x64 Edition SP2
- 1 Microsoft SQL Server 2005 Enterprise x64 Edition (per processor) SP2
- 1 Microsoft Visual Studio Standard 2005
- HP Servers include 3 years of support.

Clause 9 Related Items

Auditor's Report

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.

This implementation of the TPC Benchmark C was audited by Lorna Livingtree of Performance Metrics, Inc.

Performance Metrics, Inc.
PO Box 984
Klamath CA 95548
(phone) 707-482-0523
(fax) 707-482-0575
e-mail: lornaL@perfmetrics.com

Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™ C", the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

Requests for this TPC Benchmark C Full Disclosure Report should be sent to:

TPC
Presidio of San Francisco
Building 572B Ruger St. (surface)
P.O. Box 29920 (mail)
San Francisco, CA 94129-0920

or

Hewlett-Packard Company
Database Performance Engineering
P.O. Box 692000
Houston, TX 77269-2000



November 5, 2009

Mr. David Adams
Database Performance Engineer
Hewlett-Packard Company
20555 SH 249
Houston, TX 77070

I have verified by remote the TPC Benchmark™ C for the following configuration:

Platform: HP ProLiant DL370 G6
Database Manager: Microsoft SQL Server 2005 Enterprise X64 Edition
Operating System: Microsoft Windows Server 2008 SP2 Enterprise X64 Edition
Transaction Monitor: Microsoft COM+

System Under Test:				
CPU's	Memory	Disks (total)	90% Response	TpmC
2 Intel Xeon 2 core @ 3.2 Ghz	Main: 144 GB	1287 @ 36 GB 15 @ 72 GB 32 @ 146 GB	1.11	661,475
Clients: 16 DL360 G5				
1 Intel quad core @ 2.83 Ghz	1 GB	2 @ 72 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 54,000 warehouses, of which 53,184 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:

The 36GB disk drives used in the tested configuration are no longer orderable. The priced configuration has 72GB disk drives substituted on a one for one basis. The technical specifications for both types of drives were reviewed and the substitution is compliant with the pricing specification substitution requirements.

Sincerely,

A handwritten signature in cursive script that reads "Lorna Livingtree".

Lorna Livingtree, Certified Auditor

Appendix A: Source Code

The client source code is listed below.

dlldata.c

```
*****
DllData file -- generated by MIDL compiler

    DO NOT ALTER THIS FILE

    This file is regenerated by MIDL on every IDL file
    compile.

    To completely reconstruct this file, delete it and
    rerun MIDL
    on all the IDL files in this DLL, specifying this
    file for the
    /dlldata command line option

*****
#include <rpcproxy.h>

#ifdef __cplusplus
extern "C" {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )

PROXYFILE_LIST_START
/* Start of list */
REFERENCE_PROXY_FILE( tpcc_com_ps ),
/* End of list */
PROXYFILE_LIST_END

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

#ifdef __cplusplus
} /*extern "C" */
#endif

/* end of generated dlldata file */
```

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;
    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_WEBDDL 3
//tpcc web generated error
#define ERR_TYPE_SQL 4
//sql server generated error
#define ERR_TYPE_DBLIB 5
//dblib generated error
#define ERR_TYPE_ODBC 6
//odbc generated error
#define ERR_TYPE_SOCKET 7
//error on communication socket client rte
only
#define ERR_TYPE_DEADLOCK 8
//dblib and odbc only deadlock condition
#define ERR_TYPE_COM 9
//error from COM call
#define ERR_TYPE_TUXEDO 10
//tuxedo error
#define ERR_TYPE_OS 11
//operating system error
#define ERR_TYPE_MEMORY 12
//memory allocation error
#define ERR_TYPE_TPCC_ODBC 13
//error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB 14
//error from tpcc dblib txn module
#define ERR_TYPE_DELISRV 15
//delivery server error
#define ERR_TYPE_TXNLOG 16
//txn log error
#define ERR_TYPE_BCONN 17
//Benchcraft connection class
#define ERR_TYPE_TPCC_CONN 18
//Benchcraft connection class
#define ERR_TYPE_ENCINA 19
//Encina error
#define ERR_TYPE_COMPONENT 20
//error from COM component
#define ERR_TYPE_RTE 21
//Benchcraft rte
```

```

#define ERR_TYPE_AUTOMATION      22
    //Benchcraft automation errors
#define ERR_TYPE_DRIVER          23
    //Driver engine errors
#define ERR_TYPE_RTE_BASE       24
    //Framework errors
#define ERR_BUF_OVERFLOW         25
    //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP      26
    //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB          27
    //OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB     28
    //error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN      50
    //Benchcraft connection class
#define ERR_TYPE_TPCW_HTML      51
    //error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER      52
    //error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE  53
#define ERR_TYPE_TPCW_ENG_OS    54
#define ERR_TYPE_HTML_RESP      55
#define ERR_TYPE_TPCW_ODBC      56
#define ERR_TYPE_SCHANNEL       57
#define ERR_TYPE_THINK_LIST     58
//----- end TPC-W -----
#define ERR_TYPE_XML_PROFILE    59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN      60
    //TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE       61
    //TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE  62
    //Tpce Driver engine errors
#define ERR_TYPE_TPCE_ENG_OS    63
    //Tpce Driver engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE
    //Tpce MEE Driver engine errors

```

```

//#define ERR_TYPE_TPCE_MEE_ENG_OS
    //Tpce MEE Driver engine system errors
#define ERR_INS_MEMORY          "Insufficient Memory to continue."
#define ERR_UNKNOWN            "Unknown error."
#define ERR_MSG_BUF_SIZE       512
#define INV_ERROR_CODE         -1
#define ERR_INS_BUF_OVERFLOW   "Insufficient Buffer size to receive HTML pages."

class CBaseErr
{
public:
    enum Action
    {
        eNone = 0
    };

    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg = GetLastError(); //take the error code immediately before it is reset by other functions
    }

    if (szLoc)
    {
        m_szLoc = new char[strlen(szLoc)+1/*m_szLoc_size*/];
        strcpy(m_szLoc, szLoc);
    }
    else
        m_szLoc = NULL;

    m_szApp = new char[m_szApp_size];

    GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
}

CBaseErr(int idMsg, LPCTSTR szLoc = NULL)
{
    m_idMsg = idMsg;

    if (szLoc)
    {
        m_szLoc = new char[strlen(szLoc)+1/*m_szLoc_size*/];
        strcpy(m_szLoc, szLoc);
    }
    else
        m_szLoc = NULL;
}

```

```

    m_szApp = new char[m_szApp_size];

    GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
}

virtual ~CBaseErr(void)
{
    if (m_szApp) delete [] m_szApp;
    if (m_szLoc) delete [] m_szLoc;
};

virtual void Draw(HWND hwnd, LPCTSTR szStr = NULL)
{
    int j = 0;
    char szTmp[512];

    if (szStr)
        j = sprintf(szTmp, "%s\n", szStr);
    if (ErrorNum() != INV_ERROR_CODE)
        j += sprintf(szTmp+j, "Error = %d\n", ErrorNum());
    if (m_szLoc)
        j += sprintf(szTmp+j, "Location = %s\n", GetLocation());
    j += sprintf(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,LPCTSTR);

    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;
};

```

install.c

```

/* FILE: INSTALL.C
 * Microsoft
TPC-C Kit Ver. 4.51.000
 * Copyright
Microsoft, 2003
 * All Rights Reserved
 *
 * not audited
 *
 * PURPOSE: Automated installation
application for TPC-C Web Kit
 * Contact: Charles Levine
(clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - added COM installation
steps
 * 4.50.000 - added IIS6 configuration options
 * 4.51.000 - added routines to copy
Visual Studio runtime module (MSVCR70.DLL)
 * to
SystemRoot\System32
 */

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <tchar.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.h"
#include <process.h>

#include "resource.h"

#define WM_INITTEXT WM_USER+100

HICON hIcon;
HINSTANCE hInst;

DWORD versionExeMS;
DWORD versionExeLS;

```

```

DWORD versionExeMM;
DWORD versionDllMS;
DWORD versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA Reg;

static int iPoolThreadLimit;
static int iMaxPoolThreads;
static int iThreadTimeout;
static int iListenBackLog;
static int iAcceptExOutstanding;
static int iUriEnableCache;
static int iUriScavengerPeriod;
static int iMaxConnections;

static int iIISMajorVersion;
static int iNumberOfProcessors;

static int iMaxPhysicalMemory;
//max physical memory in MB
static char szLastFileName[64]; //
last file we worked on (for error reporting)

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT
uMsg, WPARAM wParam, LPARAM lParam);
BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT
uMsg, WPARAM wParam, LPARAM lParam);
BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam);
static void ProcessOK(HWND hwnd,
char *szDllPath, char *szWindowsPath);
static void
ReadRegistrySettings(void);
static
void
WriteRegistrySettings(char *szDllPath);
static BOOL RegisterDLL(char
*szFileName);
static int
CopyFiles(HWND hDlg, char *szDllPath, char
*szWindowsPath);
static BOOL GetInstallPath(char
*szDllPath);
static BOOL
GetWindowsInstallPath(char *szWindowsPath);
static void GetVersionInfo(char
*szDLLPath, char *szExePath);
static BOOL
CheckWWWWebService(void);
static
BOOL
StartWWWWebService(void);
static BOOL StopWWWWebService(void);
static void UpdateDialog(HWND
hDlg);
static void
ConfigureIIS6(HWND
hwnd, HWND hDlg);

SYSTEM_INFO siSysInfo;

BOOL install_com(char *szDllPath);

```

```

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

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE
hPrevInstance, LPSTR lpCmdLine, int nCmdShow )
{
int iRc;

hInst = hInstance;

InitCommonControls();

hIcon = LoadIcon(hInstance,
MAKEINTRESOURCE(IDI_ICON1));

iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG4), GetDesktopWindow(),
LicenseDlgProc);
if ( iRc )
{
iRc = DialogBox(hInstance,
MAKEINTRESOURCE(IDD_DIALOG1), GetDesktopWindow(),
MainDlgProc);
if ( iRc )
{
DialogBoxParam(hInstance,
MAKEINTRESOURCE(IDD_DIALOG2), GetDesktopWindow(),
UpdatedDlgProc, (LPARAM)iRc);
}
}

DestroyIcon(hIcon);
return 0;
}

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
HGLOBAL hRes;
HRSRC hResInfo;
BYTE *pSrc, *pDst;
DWORD dwSize;
static HFONT hFont;

switch(uMsg)
{
case WM_INITDIALOG:
hFont = CreateFont(-12,
0, 0, 0, 400, 0, 0, 0, 0, 0, 0, 0, 0, "Arial");
SendMessage(
GetDlgItem(hwnd, IDR_LICENSE1), WM_SETFONT,
(WPARAM)hFont, MAKELPARAM(0, 0) );
PostMessage(hwnd,
WM_INITTEXT, (WPARAM)0, (LPARAM)0);
return TRUE;
case WM_INITTEXT:
hResInfo =
FindResource(hInst, MAKEINTRESOURCE(IDR_LICENSE1),
"LICENSE");
dwSize =
SizeofResource(hInst, hResInfo);

```

```

        hRes =
LoadResource(hInst, hResInfo );
        pSrc = (BYTE
*)LockResource(hRes);
        pDst = (unsigned char
*)malloc(dwSize+1);
        if ( pDst )
        {
            memcpy(pDst,
pSrc, dwSize);
            pDst[dwSize]
= 0;

            SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pDst);
            free(pDst);
        }
        else
            SetDlgItemText(hwnd, IDC_LICENSE, (const
char *)pSrc);
        return TRUE;
    case WM_DESTROY:
        DeleteObject(hFont);
        return TRUE;
    case WM_COMMAND:
        if ( wParam == IDOK )
            EndDialog(hwnd, TRUE);
        if ( wParam == IDCANCEL )
            EndDialog(hwnd, FALSE);
        default:
            break;
    }
    return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    switch(uMsg)
    {
        case WM_INITDIALOG:
            switch(lParam)
            {
                case 1:
                case 2:

                    SetDlgItemText(hwnd, IDC_RESULTS, "TPC-C
Web Client Installed");

                    break;
            }
            return TRUE;
        case WM_COMMAND:
            if ( wParam == IDOK )
                EndDialog(hwnd, TRUE);
            break;
        default:
            break;
    }
}

```

```

        }
        return FALSE;
    }

    BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
    {
        PAINTSTRUCT ps;
        MEMORYSTATUS memoryStatus;
        OSVERSIONINFO VI;
        char szTmp[256];
        static char
szDllPath[256];
        static char
szWindowsPath[256];
        static char
szExePath[256];

        switch(uMsg)
        {
            case WM_INITDIALOG:
                GlobalMemoryStatus(&memoryStatus);
                iMaxPhysicalMemory =
(memoryStatus.dwTotalPhys/ 1048576);

                if (
GetWindowsInstallPath(szWindowsPath) )
                {
                    MessageBox(hwnd, "Error: Cannot determine
Windows System Root.", NULL, MB_ICONSTOP | MB_OK);

                    EndDialog(hwnd, FALSE);
                    return TRUE;
                }

                if (
GetInstallPath(szDllPath) )
                {
                    MessageBox(hwnd, "Error internet service
inetsrv is not installed.", NULL, MB_ICONSTOP |
MB_OK);

                    EndDialog(hwnd, FALSE);
                    return TRUE;
                }

                // set default values
                ZeroMemory( &Reg,
sizeof(Reg) );

                Reg.dwNumberOfDeliveryThreads = 4;
                Reg.dwMaxConnections =
100;

                Reg.dwMaxPendingDeliveries = 100;
                Reg.eDB_Protocol =
ODBC;

                Reg.eTxnMon = None;
                strcpy(Reg.szDbServer,
"");
            }
        }
    }
}

```

```

        strcpy(Reg.szDbName,
"tpcc");
        strcpy(Reg.szDbUser,
"sa");
        strcpy(Reg.szDbPassword,
"");

        iPoolThreadLimit =
iMaxPhysicalMemory * 2;
        iThreadTimeout = 86400;
        iListenBackLog = 15;
        iAcceptExOutstanding =
40;

        ReadTPCCRegistrySettings( &Reg );
        ReadRegistrySettings();

        // copy the hardware
information to the SYSTEM_INFO structure
        GetSystemInfo(&siSysInfo);
        // store the number of
processors on this system
        iNumberOfProcessors =
siSysInfo.dwNumberOfProcessors;

        GetModuleFileName(hInst, szExePath,
sizeof(szExePath));
        GetVersionInfo(szDllPath, szExePath);

        sprintf(szTmp,
"Version %d.%2d.%3d", versionExeMS, versionExeMM,
versionExeLS);
        SetDlgItemText(hwnd,
IDC_VERSION, szTmp);

        SetDlgItemText(hwnd,
IDC_PATH, szDllPath);

        SetDlgItemText(hwnd,
ED_DB_SERVER, Reg.szDbServer);
        SetDlgItemText(hwnd,
ED_DB_USER_ID, Reg.szDbUser);
        SetDlgItemText(hwnd,
ED_DB_PASSWORD, Reg.szDbPassword);
        SetDlgItemText(hwnd,
ED_DB_NAME, Reg.szDbName);

        SetDlgItemInt(hwnd,
ED_THREADS, Reg.dwNumberOfDeliveryThreads, FALSE);
        SetDlgItemInt(hwnd,
ED_MAXCONNECTION, Reg.dwMaxConnections, FALSE);
        SetDlgItemInt(hwnd,
ED_MAXDELIVERIES, Reg.dwMaxPendingDeliveries, FALSE);
        SetDlgItemInt(hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, iPoolThreadLimit,
FALSE);
        SetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, iThreadTimeout, FALSE);
    }
}

```

```

        SetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, iListenBackLog, FALSE);
        SetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

        // check OS version
level for COM. Must be at least Windows 2000
        VI.dwOSVersionInfoSize
= sizeof(VI);
        GetVersionEx( &VI );
        if (VI.dwMajorVersion <
5)
        {
                HWND hDlg =
GetDlgItem( hwnd, IDC_TM_MTS );
                EnableWindow(
hDlg, 0 ); // disable COM option
                if
(Reg.eTxnMon == COM)
                        Reg.eTxnMon = None;
        }
        CheckDlgButton(hwnd,
IDC_TM_NONE, 0);
        CheckDlgButton(hwnd,
IDC_TM_MTS, 0);
        switch (Reg.eTxnMon)
        {
        case None:
                CheckDlgButton(hwnd, IDC_TM_NONE, 1);
                break;
        case COM:
                CheckDlgButton(hwnd, IDC_TM_MTS, 1);
                break;
        }
        return TRUE;
        case WM_PAINT:
                if ( IsIconic(hwnd) )
                {
                        BeginPaint(hwnd, &ps);
                        DrawIcon(ps.hdc, 0, 0, hIcon);
                        EndPaint(hwnd, &ps);
                        return TRUE;
                }
                break;
        case WM_COMMAND:
                if ( HIWORD(wParam) ==
BN_CLICKED )
                {
                        LOWORD(wParam) )
                                switch(
                                {
                                        case IDOK:

```

```

ProcessOK(hwnd, szDllPath, szWindowsPath);
return TRUE;
case IDCANCEL:
EndDialog(hwnd, FALSE);
return TRUE;
default:
return FALSE;
}
}
static void ProcessOK(HWND hwnd, char *szDllPath,
char *szWindowsPath)
{
        int d;
        HWND hDlg;
        int rc;
        BOOL bSvcRunning;
        char szFullName[256];
        char szErrTxt[128];
        // Check whether Service Pack 1 has been
installed if
// running on Windows Server 2003. The RTM
version has
// a limitation on the number of concurrent
HTTP connections.
//
OSVERSIONINFOEX VersionInfo;
        VersionInfo.dwOSVersionInfoSize =
sizeof(OSVERSIONINFOEX);
        if
(GetVersionEx((LPOSVERSIONINFO)&VersionInfo))
        {
                if (VersionInfo.dwMajorVersion ==
5 && // Windows 2000/2003 Server?
                VersionInfo.dwMinorVersion == 2 && //
Windows 2003 Server?
                VersionInfo.wServicePackMajor == 0) //
Service Pack installed?
                {
                        TCHAR szMsg[256];
                        _sntprintf(szMsg,
sizeof(szMsg),

```

```

"Warning:
running on Windows Server 2003 without at least
Service Pack 1\n"
"limits the
number of concurrent HTTP connections to around
8000.");
        MessageBox(hwnd, szMsg,
_T("Service Pack not Installed"), MB_ICONEXCLAMATION
| MB_OK);
}
}
// read settings from dialog
Reg.dwNumberOfDeliveryThreads =
GetDlgItemInt(hwnd, ED_THREADS, &d, FALSE);
Reg.dwMaxConnections = GetDlgItemInt(hwnd,
ED_MAXCONNECTION, &d, FALSE);
Reg.dwMaxPendingDeliveries =
GetDlgItemInt(hwnd, ED_MAXDELIVERIES, &d, FALSE);
        GetDlgItemText(hwnd, ED_DB_SERVER,
Reg.szDbServer, sizeof(Reg.szDbServer));
        GetDlgItemText(hwnd, ED_DB_USER_ID,
Reg.szDbUser, sizeof(Reg.szDbUser));
        GetDlgItemText(hwnd, ED_DB_PASSWORD,
Reg.szDbPassword, sizeof(Reg.szDbPassword));
        GetDlgItemText(hwnd, ED_DB_NAME,
Reg.szDbName, sizeof(Reg.szDbName));
        if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE)
)
                Reg.eTxnMon = None;
        else if ( IsDlgButtonChecked(hwnd,
IDC_TM_MTS) )
                Reg.eTxnMon = COM;
        iPoolThreadLimit = GetDlgItemInt(hwnd,
ED_IIS_MAX_THREAD_POOL_LIMIT, &d, FALSE);
        iThreadTimeout = GetDlgItemInt(hwnd,
ED_IIS_THREAD_TIMEOUT, &d, FALSE);
        iListenBackLog = GetDlgItemInt(hwnd,
ED_IIS_LISTEN_BACKLOG, &d, FALSE);
        iAcceptExOutstanding = GetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE);
        ShowWindow(hwnd, SW_HIDE);
        hDlg = CreateDialog(hInst,
MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
        ShowWindow(hDlg, SW_SHOWNA);
        UpdateDialog(hDlg);
        // check to see if the web services are
running
        bSvcRunning = CheckWWWWebService();
        if ( bSvcRunning )
        {
                SetDlgItemText(hDlg, IDC_STATUS,
"Stopping Web Service.");
                SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
                UpdateDialog(hDlg);

```

```

        StopWWWService();
        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);
    }

    // write binaries to inetpub\wwwroot
    rc = CopyFiles(hDlg, szDllPath,
szWindowsPath);
    if (!rc)
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "Error(s)
occured when creating " );
        strcat( szErrTxt, szLastFileName
);
        MessageBox(hwnd, szErrTxt, NULL,
MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }

    // while we have the web services shutdown,
check to see if this
    // is IIS6. If it is, then call
ConfigureIIS6
    if ( iIISMajorVersion == 6)
    {
        ConfigureIIS6(hwnd, hDlg);
    }

    //if we stopped service restart it.
    if ( bSvcRunning )
    {
        SetDlgItemText(hDlg, IDC_STATUS,
"Starting Web Service.");
        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);
        StartWWWService();
    }

    // update registry
    SetDlgItemText(hDlg, IDC_STATUS, "Updating
Registry.");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);
    WriteRegistrySettings(szDllPath);

    // register com proxy stub
    strcpy(szFullName, szDllPath);
    strcat(szFullName, "tpcc_com_ps.dll");
    if (!RegisterDLL(szFullName))
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "Error occured
when registering " );
        strcat( szErrTxt, szFullName );
        MessageBox(hwnd, szErrTxt, NULL,
MB_ICONSTOP | MB_OK);

```

```

        EndDialog(hwnd, 0);
        return;
    }

    // if using COM
    if (Reg.eTxnMon == COM)
    {
        SetDlgItemText(hDlg, IDC_STATUS,
"Configuring COM.");
        SendDlgItemMessage(hDlg,
IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        if (install_com(szDllPath))
        {
            ShowWindow(hwnd,
SW_SHOWNA);
            DestroyWindow(hDlg);
            strcpy( szErrTxt,
"Error occured when configuring COM settings." );
            MessageBox(hwnd,
szErrTxt, NULL, MB_ICONSTOP | MB_OK);
            EndDialog(hwnd, 0);
            return;
        }
    }

    Sleep(100);

    ShowWindow(hwnd, SW_SHOWNA);
    DestroyWindow(hDlg);

    EndDialog(hwnd, rc);
    return;
}

static void ReadRegistrySettings(void)
{
    HKEY    hKey;
    DWORD  size;
    DWORD  type;

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\InetStp", 0, KEY_READ, &hKey)
== ERROR_SUCCESS )
    {
        size = sizeof(iIISMajorVersion);
        if ( RegQueryValueEx(hKey,
"MajorVersion", 0, &type, (char *)&iIISMajorVersion,
&size) == ERROR_SUCCESS )
            if ( !iIISMajorVersion
)
                iIISMajorVersion = 5;
    }

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Param
eters", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
    {
        if ( iIISMajorVersion == 6)
            {

```

```

        // since IIS6 handles
the pool thread parameters differently, we need to
fill in the dialog

        // with the
MaxPoolThreads rather than PoolThreadLimit
        // for ease of coding,
we are just going to stuff the value into
iPoolThreadLimit
        size = sizeof(iPoolThreadLimit);
        if (
RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type,
(char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
            if ( !iPoolThreadLimit
)
                iPoolThreadLimit = iMaxPhysicalMemory * 2;
            else
            {
                size =
sizeof(iPoolThreadLimit);
                if (
RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type,
(char *)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                    if ( !iPoolThreadLimit
)
                        iPoolThreadLimit = iMaxPhysicalMemory * 2;
            }

        size = sizeof(iThreadTimeout);
        if ( RegQueryValueEx(hKey,
"ThreadTimeout", 0, &type, (char *)&iThreadTimeout,
&size) == ERROR_SUCCESS )
            if ( !iThreadTimeout )
                iThreadTimeout = 86400;

        size = sizeof(iListenBackLog);
        if ( RegQueryValueEx(hKey,
"ListenBackLog", 0, &type, (char *)&iListenBackLog,
&size) == ERROR_SUCCESS )
            if ( !iListenBackLog )
                iListenBackLog = 15;

        RegCloseKey(hKey);
    }

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Paramete
rs", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
    {
        size =
sizeof(iAcceptExOutstanding);
        if ( RegQueryValueEx(hKey,
"AcceptExOutstanding", 0, &type, (char
*)&iAcceptExOutstanding, &size) == ERROR_SUCCESS )
            if (
!iAcceptExOutstanding )
                iAcceptExOutstanding = 40;
    }

```

```

        RegCloseKey(hKey);
    }
    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\HTTP\\Parameters", 0, KEY_READ, &hKey) == ERROR_SUCCESS )
    {
        size = sizeof(iUriEnableCache);
        if ( RegQueryValueEx(hKey,
"UriEnableCache", 0, &type, (char *)&iUriEnableCache,
&size) == ERROR_SUCCESS )
            if ( !iUriEnableCache )
                iUriEnableCache = 0;

        size =
sizeof(iUriScavengerPeriod);
        if ( RegQueryValueEx(hKey,
"UriScavengerPeriod", 0, &type, (char
*)&iUriScavengerPeriod, &size) == ERROR_SUCCESS )
            if (
!iUriScavengerPeriod )
                iUriScavengerPeriod = 10800;

        size = sizeof(iMaxConnections);
        if ( RegQueryValueEx(hKey,
"MaxConnections", 0, &type, (char *)&iMaxConnections,
&size) == ERROR_SUCCESS )
            if ( !iMaxConnections )
                iMaxConnections = 100000;

        RegCloseKey(hKey);
    }
}

static void WriteRegistrySettings(char *szDllPath)
{
    HKEY    hKey;
    DWORD   dwDisposition;
    char    szTmp[256];
    char    *ptr;
    int     iRc;

    if ( RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\\Microsoft\\TPCC", 0, NULL,
REG_OPTION_NON_VOLATILE, KEY_ALL_ACCESS, NULL, &hKey,
&dwDisposition) == ERROR_SUCCESS )
    {
        strcpy(szTmp, szDllPath);
        ptr = strstr(szTmp, "tpcc");
        if ( ptr )
            *ptr = 0;

        RegSetValueEx(hKey, "Path", 0,
REG_SZ, szTmp, strlen(szTmp)+1);

        RegSetValueEx(hKey,
"NumberOfDeliveryThreads", 0, REG_DWORD, (char
*)&Reg.dwNumberOfDeliveryThreads,
sizeof(Reg.dwNumberOfDeliveryThreads));
    }
}

```

```

        RegSetValueEx(hKey,
"MaxConnections", 0, REG_DWORD, (char
*)&Reg.dwMaxConnections,
sizeof(Reg.dwMaxConnections));
        RegSetValueEx(hKey,
"MaxPendingDeliveries", 0, REG_DWORD, (char
*)&Reg.dwMaxPendingDeliveries,
sizeof(Reg.dwMaxPendingDeliveries));

        RegSetValueEx(hKey,
"DB Protocol", 0, REG_SZ,
szDBNames[Reg.eDB_Protocol],
strlen(szDBNames[Reg.eDB_Protocol])+1);
        RegSetValueEx(hKey, "TxnMonitor",
0, REG_SZ, szTxnMonNames[Reg.eTxnMon],
strlen(szTxnMonNames[Reg.eTxnMon])+1);

        RegSetValueEx(hKey, "DbServer",
0, REG_SZ, Reg.szDbServer, strlen(Reg.szDbServer)+1);
        RegSetValueEx(hKey, "DbName", 0,
REG_SZ, Reg.szDbName, strlen(Reg.szDbName)+1);
        RegSetValueEx(hKey, "DbUser", 0,
REG_SZ, Reg.szDbUser, strlen(Reg.szDbUser)+1);
        RegSetValueEx(hKey, "DbPassword",
0, REG_SZ, Reg.szDbPassword,
strlen(Reg.szDbPassword)+1);

        strcpy(szTmp, "YES");
        RegSetValueEx(hKey,
"COM_SinglePool", 0, REG_SZ, szTmp, strlen(szTmp)+1);

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    if (
(iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Param
eters", 0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        // if this is IIS6, then we need
to treat the PoolThreadLimit differently
        // if IIS6, then PoolThreadLimit
is the maximum number of threads for the entire
system.
        // IIS6 added MaxPoolThreads
which controls the number of threads per processor.
For IIS6
        // we will set MaxPoolThreads to
the value the user provided in the dialog and then
set
        // PoolThreadLimit to
MaxPoolThreads * number of processors on this system
        if ( iIISMajorVersion == 6 )
        {
            iMaxPoolThreads =
iPoolThreadLimit;
            iPoolThreadLimit =
iMaxPoolThreads * iNumberOfProcessors;
        }
    }
}

```

```

        RegSetValueEx(hKey,
"PoolThreadLimit", 0, REG_DWORD, (char
*)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
        RegSetValueEx(hKey,
"MaxPoolThreads", 0, REG_DWORD, (char
*)&iMaxPoolThreads, sizeof(iMaxPoolThreads));
    }
    else
    {
        RegSetValueEx(hKey,
"PoolThreadLimit", 0, REG_DWORD, (char
*)&iPoolThreadLimit, sizeof(iPoolThreadLimit));
    }

    RegSetValueEx(hKey,
"ThreadTimeout", 0, REG_DWORD, (char
*)&iThreadTimeout, sizeof(iThreadTimeout));
    RegSetValueEx(hKey,
"ListenBackLog", 0, REG_DWORD, (char
*)&iListenBackLog, sizeof(iListenBackLog));

    RegFlushKey(hKey);
    RegCloseKey(hKey);
}

    if (
(iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Paramete
rs", 0, NULL, REG_OPTION_NON_VOLATILE,
KEY_ALL_ACCESS, NULL, &hKey, &dwDisposition)) ==
ERROR_SUCCESS )
    {
        RegSetValueEx(hKey,
"AcceptExOutstanding", 0, REG_DWORD, (char
*)&iAcceptExOutstanding,
sizeof(iAcceptExOutstanding));

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

    return;
}

BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg,
WPARAM wParam, LPARAM lParam)
{
    if ( uMsg == WM_INITDIALOG )
    {
        SendDlgItemMessage(hwnd,
IDC_PROGRESS1, PBM_SETRANGE, 0, MAKELPARAM(0, 13));
        SendDlgItemMessage(hwnd,
IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1, 0);
        return TRUE;
    }
    return FALSE;
}

BOOL RegisterDLL(char *szFileName)
{
    HINSTANCE hLib;
    FARPROC   lpDllEntryPoint;
}

```

```

        hLib = LoadLibrary(szFileName);
        if ( hLib == NULL )
            return FALSE;
        // Find the entry point.
        lpDllEntryPoint = GetProcAddress(hLib,
"DllRegisterServer");
        if (lpDllEntryPoint != NULL)
        {
            return ((*lpDllEntryPoint)() ==
S_OK);
        }
        else
            return FALSE; //unable to
locate entry point
}

BOOL FileFromResource( char *szResourceName, int
iResourceId, char *szDllPath, char *szFileName )
{
    HGLOBAL          hDLL;
    HRSRC            hResInfo;
    HANDLE           hFile;
    DWORD            dwSize;
    BYTE             *pSrc;
    DWORD            d;
    char             szFullName[256];

    hResInfo = FindResource(hInst,
MAKEINTRESOURCE(iResourceId), szResourceName);

    strcpy(szFullName, szDllPath);
    strcat(szFullName, szFileName);

    dwSize = SizeofResource(hInst, hResInfo);
    hDLL = LoadResource(hInst, hResInfo);
    pSrc = (BYTE *)LockResource(hDLL);
    remove(szFullName);

    if ( !(hFile = CreateFile(szFullName,
GENERIC_WRITE, 0, NULL, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, NULL)) )
        return FALSE;

    if ( !WriteFile(hFile, pSrc, dwSize, &d,
NULL) )
        return FALSE;

    CloseHandle(hFile);

    UnlockResource(hDLL);
    FreeResource(hDLL);
    return TRUE;
}

static int CopyFiles(HWND hDlg, char *szDllPath, char
*szWindowsPath)
{
    SetDlgItemText(hDlg, IDC_STATUS, "Copying
Files...");
    SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

```

```

        // install TPCC.DLL
        strcpy( szLastFileName, "tpcc.dll" );
        if (!FileFromResource( "TPCCDLL",
IDR_TPCCDLL, szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install MSVCR71.DLL
        strcpy( szLastFileName, "msvcr71.dll" );
        if (!FileFromResource( "MSVCR71",
IDR_MSVCR71, szWindowsPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_odbc.dll
        strcpy( szLastFileName, "tpcc_odbc.dll" );
        if (!FileFromResource( "ODBC_DLL",
IDR_ODBC_DLL, szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com.dll
        strcpy( szLastFileName, "tpcc_com.dll" );
        if (!FileFromResource( "COM_DLL",
IDR_COM_DLL, szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com_all.tlb
        strcpy( szLastFileName, "tpcc_com_all.tlb"
);
        if (!FileFromResource( "COM_TYPLIB",
IDR_COMTYPLIB_DLL, szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com_ps.dll
        strcpy( szLastFileName, "tpcc_com_ps.dll"
);
        if (!FileFromResource( "COM_PS_DLL",
IDR_COMPS_DLL, szDllPath, szLastFileName ))
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        // install tpcc_com_all.dll
        strcpy( szLastFileName, "tpcc_com_all.dll"
);
        if (!FileFromResource( "COM_ALL_DLL",
IDR_COMALL_DLL, szDllPath, szLastFileName ))

```

```

        return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        SendDlgItemMessage(hDlg, IDC_PROGRESS1,
PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

        return 1;
    }

static BOOL GetInstallPath(char *szDllPath)
{
    HKEY    hKey;
    BYTE    szData[256];
    DWORD   sv;
    BOOL    bRc;
    int     len;
    int     iRc;

    // Registry key
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InetStp\PathWWW
Root is used to find the
    // IIS default web site directory and
determine that IIS is installed.

    szDllPath[0] = 0;
    bRc = TRUE;
    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\Microsoft\InetStp", 0, KEY_ALL_ACCESS,
&hKey) == ERROR_SUCCESS )
    {
        sv = sizeof(szData);
        iRc = RegQueryValueEx( hKey,
"PathWWWRoot", NULL, NULL, szData, &sv ); // used by
IIS 5.0 & 6.0
        if (iRc == ERROR_SUCCESS)
        {
            bRc = FALSE;
            strcpy(szDllPath,
szData);
            len =
strlen(szDllPath);
            if ( szDllPath[len-1]
!= '\\' )
            {
                szDllPath[len] = '\\';
                szDllPath[len+1] = 0;
            }
            RegCloseKey(hKey);
        }
        return bRc;
    }

static BOOL GetWindowsInstallPath(char
*szWindowsPath)

```

```

{
    HKEY hKey;
    BYTE szData[256];
    DWORD sv;
    BOOL bRc;
    int len;
    int iRc;

    // Registry key
    HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
    NT\CurrentVersion\SystemRoot is used to find the
    // system root to install the VC70 DLL.

    szWindowsPath[0] = 0;
    bRc = TRUE;
    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE\Microsoft\Windows NT\CurrentVersion", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
    {
        sv = sizeof(szData);
        iRc = RegQueryValueEx( hKey,
"SystemRoot", NULL, NULL, szData, &sv );
        if (iRc == ERROR_SUCCESS)
        {
            bRc = FALSE;
            strcpy(szWindowsPath,
szData);
            len =
strlen(szWindowsPath);
            if ( szWindowsPath[len-
1] != '\\' )
            {
                szWindowsPath[len] = '\\';
                szWindowsPath[len+1] = 0;
            }
            // now append the path
            strcat(szWindowsPath,
"SYSTEM32\");
        }

        RegCloseKey(hKey);
    }

    return bRc;
}

static void GetVersionInfo(char *szDLLPath, char
*szExePath)
{
    DWORD d;
    DWORD dwSize;
    DWORD dwBytes;
    char
*ptr;
    VS_FIXEDFILEINFO *vs;

    versionDllMS = 0;
    versionDllLS = 0;

```

```

        if ( _access(szDLLPath, 00) == 0 )
        {
            dwSize =
GetFileVersionInfoSize(szDLLPath, &d);
            if ( dwSize )
            {
                ptr = (char
*)malloc(dwSize);

                GetFileVersionInfo(szDLLPath, 0, dwSize,
ptr);
                VerQueryValue(ptr,
"\\",&vs, &dwBytes);

                >dwProductVersionMS;
                versionDllMS = vs-
                >dwProductVersionLS;
                versionDllLS = vs-
                free(ptr);
            }
        }

        versionExeMS = 0x7FFF;
        versionExeLS = 0x7FFF;
        dwSize = GetFileVersionInfoSize(szExePath,
&d);
        if ( dwSize )
        {
            ptr = (char *)malloc(dwSize);
            GetFileVersionInfo(szExePath, 0,
dwSize, ptr);
            VerQueryValue(ptr, "\\",&vs,
&dwBytes);

            versionExeMS = vs-
            >dwProductVersionMS;
            versionExeLS = LOWORD(vs-
            >dwProductVersionLS);
            versionExeMM = HIWORD(vs-
            >dwProductVersionLS);
            free(ptr);
        }
        return;
    }

    static BOOL CheckWWWWebService(void)
    {
        SC_HANDLE schSCManager;
        SC_HANDLE schService;
        SERVICE_STATUS ssStatus;

        schSCManager = OpenSCManager(NULL, NULL,
SC_MANAGER_ALL_ACCESS);
        schService = OpenService(schSCManager,
TEXT("W3SVC"), SERVICE_ALL_ACCESS);
        if (schService == NULL)
            return FALSE;

        if (! QueryServiceStatus(schService,
&ssStatus) )
            goto ServiceNotRunning;

        if ( !ControlService(schService,
SERVICE_CONTROL_STOP, &ssStatus) )

```

```

            goto ServiceNotRunning;
        //start Service pending, Check the status
        until the service is running.
        if (! QueryServiceStatus(schService,
&ssStatus) )
            goto ServiceNotRunning;

        CloseServiceHandle(schService);
        return TRUE;

    ServiceNotRunning:
        CloseServiceHandle(schService);
        return FALSE;
}

static BOOL StartWWWWebService(void)
{
    SC_HANDLE schSCManager;
    SC_HANDLE schService;
    SERVICE_STATUS ssStatus;
    DWORD dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL,
SC_MANAGER_ALL_ACCESS);
    schService = OpenService(schSCManager,
TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! StartService(schService, 0, NULL) )
        goto StartWWWWebErr;
    //start Service pending, Check the status
    until the service is running.
    if (! QueryServiceStatus(schService,
&ssStatus) )
        goto StartWWWWebErr;
    while( ssStatus.dwCurrentState !=
SERVICE_RUNNING)
    {
        dwOldCheckPoint =
ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);

        //Wait for the specified interval.
        if (
!QueryServiceStatus(schService, &ssStatus) )
            //Check the status again.
            break;
        if (dwOldCheckPoint >=
ssStatus.dwCheckPoint) //Break if
        the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState ==
SERVICE_RUNNING)
        goto StartWWWWebErr;

    CloseServiceHandle(schService);

```

```

        return TRUE;
StartWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static BOOL StopWWWebService(void)
{
    SC_HANDLE          schSCManager;
    SC_HANDLE          schService;
    SERVICE_STATUS     ssStatus;
    DWORD              dwOldCheckPoint;

    schSCManager = OpenSCManager(NULL, NULL,
    SC_MANAGER_ALL_ACCESS);
    //schService = OpenService(schSCManager,
    TEXT("W3SVC"), SERVICE_ALL_ACCESS);
    schService = OpenService(schSCManager,
    TEXT("IISADMIN"), SERVICE_ALL_ACCESS);
    if (schService == NULL)
        return FALSE;

    if (! QueryServiceStatus(schService,
    &ssStatus) )
        goto StopWWWebErr;

    if ( !ControlService(schService,
    SERVICE_CONTROL_STOP, &ssStatus) )
        goto StopWWWebErr;
    //start Service pending, Check the status
    until the service is running.
    if (! QueryServiceStatus(schService,
    &ssStatus) )
        goto StopWWWebErr;
    while( ssStatus.dwCurrentState ==
    SERVICE_RUNNING)
    {
        dwOldCheckPoint =
    ssStatus.dwCheckPoint;
        //Save the current checkpoint.
        Sleep(ssStatus.dwWaitHint);

        //Wait for the specified interval.
        if (
    !QueryServiceStatus(schService, &ssStatus) )
            //Check the status again.
            break;
        if (dwOldCheckPoint >=
    ssStatus.dwCheckPoint) //Break if
    the checkpoint has not been incremented.
            break;
    }

    if (ssStatus.dwCurrentState ==
    SERVICE_RUNNING)
        goto StopWWWebErr;

    CloseServiceHandle(schService);
    return TRUE;
}

```

```

StopWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
    MSG msg;

    UpdateWindow(hDlg);
    while( PeekMessage(&msg, hDlg, 0, 0,
    PM_REMOVE) )
    {
        TranslateMessage(&msg);
        DispatchMessage(&msg);
    }
    Sleep(250);
    return;
}

static void ConfigureIIS6(HWND hwnd, HWND hDlg)
{
    int         irc;
    char        szErrTxt[128];
    FILE        *fErrorFile;

    SetDlgItemText(hDlg, IDC_STATUS,
    "Configuring IIS6...");
    //SendDlgItemMessage(hDlg, IDC_PROGRESS1,
    PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

    irc = system("IIS6_CONFIG.CMD");

    // since the return code from the command
    file is always 1,
    // check to see if the file iis6_config.err
    exists
    // if it does, then something hosed
    fErrorFile = fopen("IIS6_CONFIG.err","r");
    if ( fErrorFile != NULL )
    {
        ShowWindow(hwnd, SW_SHOWNA);
        DestroyWindow(hDlg);
        strcpy( szErrTxt, "IIS6
    configuration error." );
        strcat( szErrTxt, "Check
    iis6_config.err" );
        MessageBox(hwnd, szErrTxt, NULL,
    MB_ICONSTOP | MB_OK);
        EndDialog(hwnd, 0);
        return;
    }
}

install.h
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//

```

```

#define IDD_DIALOG1          101
#define IDI_ICON1           102
#define IDR_TPCCDLL         103
#define IDD_DIALOG2        105
#define IDI_ICON2           106
#define IDR_DELIVERY        107
#define IDD_DIALOG3        108

#define BN_LOG               1001
#define ED_KEEP              1002
#define ED_THREADS          1003
#define ED_THREADS2         1004
#define IDC_PATH             1007
#define IDC_VERSION         1009
#define IDC_RESULTS         1010
#define IDC_PROGRESS1       1011
#define IDC_STATUS          1012
#define IDC_BUTTON1         1013
#define ED_MAXCONNECTION    1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_ODBC             1022
#define IDC_CONNECT_POOL    1023
#define ED_USER_CONNECT_DELAY_TIME 1024

// Next default values for new objects
//
install.rc
// Microsoft Visual C++ 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

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

IDD_DIALOG1 DIALOGEX 0, 0, 219, 324
STYLE DS_SETFONT | DS_MODALFRAME | DS_CENTER |
WS_MINIMIZEBOX | WS_POPUP |
WS_CAPTION | WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif", 0, 0, 0x1
BEGIN
    EDITTEXT          ED_THREADS,164,45,34,12,ES_RIGHT
    | ES_NUMBER,
    WS_EX_RTLREADING

    EDITTEXT
ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
WS_EX_RTLREADING

    EDITTEXT
ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
WS_EX_RTLREADING

    CONTROL
"None", IDC_TM_NONE, "Button", BS_AUTORADIOBUTTON |
WS_GROUP |
WS_TABSTOP, 43, 104, 33, 10
    CONTROL
"COM", IDC_TM_MTS, "Button", BS_AUTORADIOBUTTON |
WS_TABSTOP, 94, 104, 32, 10

    EDITTEXT
ED_DB_SERVER, 131, 145, 67, 12, ES_AUTOHSCROLL
    EDITTEXT
ED_DB_USER_ID, 131, 158, 67, 12, ES_AUTOHSCROLL
    EDITTEXT
ED_DB_PASSWORD, 131, 171, 67, 12, ES_AUTOHSCROLL
    EDITTEXT
ED_DB_NAME, 131, 184, 67, 12, ES_AUTOHSCROLL
    EDITTEXT
ED_IIS_MAX_THREAD_POOL_LIMIT, 164, 226, 34, 12, ES_RIGHT |
ES_NUMBER, WS_EX_RTLREADING

    EDITTEXT
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, 164, 240, 34, 12, ES_RI
GHT |
ES_NUMBER, WS_EX_RTLREADING

    EDITTEXT
ED_IIS_THREAD_TIMEOUT, 164, 254, 34, 12, ES_RIGHT |
ES_NUMBER,
WS_EX_RTLREADING

```

```

EDITTEXT
ED_IIS_LISTEN_BACKLOG, 164, 268, 34, 12, ES_RIGHT |
ES_NUMBER,
WS_EX_RTLREADING
DEFPUSHBUTTON "OK", IDOK, 53, 296, 50, 14
PUSHBUTTON "Cancel", IDCANCEL, 119, 296, 50, 14
EDITTEXT
IDC_PATH, 106, 26, 91, 13, ES_AUTOHSCROLL | ES_READONLY
LTEXT "Number of Delivery
Threads:", IDC_STATIC, 35, 45, 115, 12
LTEXT "Max Number of
Connections:", IDC_STATIC, 35, 73, 115, 12
RTEXT "Version
4.11", IDC_VERSION, 120, 4, 89, 9
LTEXT "IIS Max Thread Pool
Limit:", IDC_STATIC, 36, 226, 115, 12
LTEXT "Web Service Backlog Queue
Size:", IDC_STATIC, 36, 240, 115,
12
LTEXT "IIS Thread Timeout
(seconds):", IDC_STATIC, 36, 254, 115, 12
LTEXT "IIS Listen
Backlog:", IDC_STATIC, 36, 270, 115, 10
LTEXT "Installation
directory:", IDC_STATIC, 35, 29, 71, 10
GROUPBOX "Transaction
Monitor", IDC_STATIC, 33, 90, 165, 33
LTEXT "Server
Name:", IDC_STATIC, 35, 148, 56, 8
LTEXT "User ID:", IDC_STATIC, 35, 161, 60, 8
LTEXT "User
Password:", IDC_STATIC, 35, 174, 83, 8
LTEXT "Database
Name:", IDC_STATIC, 35, 187, 54, 8
GROUPBOX "SQL Server Connection
Properties", IDC_STATIC, 22, 132, 187,
74
GROUPBOX "Web Client
Properties", IDC_STATIC, 22, 15, 187, 113
GROUPBOX "IIS
Settings", IDC_STATIC, 22, 210, 187, 79
LTEXT "Max Pending
Deliveries:", IDC_STATIC, 35, 59, 115, 12
END

IDD_DIALOG2 DIALOGEX 0, 0, 117, 62
STYLE DS_SETFONT | DS_SETFOREGROUND | DS_3DLOOK |
DS_CENTER | WS_POPUP |
WS_BORDER
EXSTYLE WS_EX_STATICEDGE
FONT 12, "MS Sans Serif", 0, 0, 0x1
BEGIN
DEFPUSHBUTTON "OK", IDOK, 33, 45, 50, 9
CTEXT "HTML TPC-C Installation
Successful", IDC_RESULTS, 7, 22,
102, 18, 0, WS_EX_CLIENTEDGE
ICON
IDI_ICON2, IDC_STATIC, 50, 7, 18, 20, SS_REALSIZEIMAGE,
WS_EX_TRANSPARENT
END

IDD_DIALOG3 DIALOG 0, 0, 91, 40

```

```

STYLE DS_SYSMODAL | DS_SETFONT | DS_MODALFRAME |
DS_3DLOOK | DS_CENTER |
WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
    CONTROL
"Progress1", IDC_PROGRESS1, "msctls_progress32", WS_BORD
ER,
7, 20, 77, 13

    CTEXT
"Static", IDC_STATUS, 7, 7, 77, 12, SS_SUNKEN
END

IDD_DIALOG4 DIALOG 0, 0, 291, 202
STYLE DS_SETFONT | DS_MODALFRAME | DS_CENTER |
WS_POPUP | WS_CAPTION |
WS_SYSMENU
CAPTION "Client End User License"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT
IDC_LICENSE, 7, 7, 271, 167, ES_MULTILINE | ES_AUTOVSCROLL
|
ES_AUTOHSCROLL | ES_READONLY |
WS_VSCROLL | WS_HSCROLL
DEFPUSHBUTTON "I &Agree", IDOK, 87, 181, 50, 14
PUSHBUTTON "&Cancel", IDCANCEL, 153, 181, 50, 14
END

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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 22
        RIGHTMARGIN, 209
        VERTGUIDE, 35
        VERTGUIDE, 198
        TOPMARGIN, 4
        BOTTOMMARGIN, 318
    END

    IDD_DIALOG2, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 109
        TOPMARGIN, 7
        BOTTOMMARGIN, 54
    END

    IDD_DIALOG3, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 84
        TOPMARGIN, 7
    END

```

```

        BOTTOMMARGIN, 33
    END

    IDD_DIALOG4, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 278
        TOPMARGIN, 7
        BOTTOMMARGIN, 195
    END
END
#endif // APSTUDIO_INVOKED

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
////////////////////////////////////
//
// TEXTINCLUDE
//
1 TEXTINCLUDE
BEGIN
    "resource.h\0"
END

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

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

#endif // APSTUDIO_INVOKED

////////////////////////////////////
////////////////////////////////////
//
// Icon
//
// Icon with lowest ID value placed first to ensure
// remains consistent on all systems.
IDI_ICON1            ICON
"icon1.ico"
IDI_ICON2            ICON
"icon2.ico"

////////////////////////////////////
////////////////////////////////////
//
// TPCCDLL
//
IDR_TPCDDLL          TPCDDLL
"..\\..\\isapi_dll\\bin\\tpcc.dll"

```

```

////////////////////////////////////
////////////////////////////////////
//
// Version
//
VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,50,0
PRODUCTVERSION 0,4,50,0
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
    FILEFLAGS 0x1L
#else
    FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x1L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C Web Client"
        END
    END
    "Installer"
        VALUE "CompanyName", "Microsoft"
        VALUE "FileDescription", "install"
        VALUE "FileVersion", "0, 4, 20, 0"
        VALUE "InternalName", "install"
        VALUE "LegalCopyright", "Copyright ©
1999"
        VALUE "OriginalFilename", "install.exe"
        VALUE "ProductName", "Microsoft install"
        VALUE "ProductVersion", "0, 4, 20, 0"
    END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

////////////////////////////////////
////////////////////////////////////
//
// LICENSE
//
IDR_LICENSE1        LICENSE
"license.txt"

////////////////////////////////////
////////////////////////////////////
//
// ODBC_DLL
//
IDR_ODBC_DLL        ODBC_DLL
"..\\..\\db_odbc_dll\\bin\\Release\\tpcc_odbc.dll"

```

```

////////////////////////////////////
////////////////////////////////////
//
// COM_DLL
//
IDR_COM_DLL          COM_DLL
"..\\..\\tm_com_dll\\bin\\tpcc_com.dll"

////////////////////////////////////
////////////////////////////////////
//
// COM_PS_DLL
//
IDR_COMPS_DLL        COM_PS_DLL
"..\\..\\tpcc_com_ps\\bin\\tpcc_com_ps.dll"

////////////////////////////////////
////////////////////////////////////
//
// COM_ALL_DLL
//
IDR_COMALL_DLL        COM_ALL_DLL
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"

////////////////////////////////////
////////////////////////////////////
//
// COM_TYPLIB
//
IDR_COMTYPLIB_DLL    COM_TYPLIB
"..\\..\\tpcc_com_all\\src\\tpcc_com_all.tlb"

////////////////////////////////////
////////////////////////////////////
//
// MSVCRT71
//
IDR_MSVCRT71         MSVCRT71
"C:\\WINDOWS\\system32\\msvcr71.dll"
#endif // English (U.S.) resources
////////////////////////////////////
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
////////////////////////////////////
////////////////////////////////////
//
// not APSTUDIO_INVOKED
//
#endif // not APSTUDIO_INVOKED

```

install_com.cpp

```
/* FILE: INSTALL_COM.CPP
 * Microsoft
 * TPC-C Kit Ver. 4.51.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 *
 * not audited
 *
 * PURPOSE: installation code for COM
 * application for TPC-C Web Kit
 * Contact: Charles Levine
 * (clevine@microsoft.com)
 *
 * Change history:
 * 4.20.000 - first version
 */

#define WIN32_WINNT 0x0500

#include <comdef.h>
#include <comadmin.h>
#include <stdio.h>
#include <getchar.h>

extern "C"
{
    BOOL install_com(char *szDllPath);
}

BOOL install_com(char *szDllPath)
{
    ICOMAdminCatalog* pCOMAdminCat = NULL;
    ICatalogCollection* pCatalogCollectionApp
= NULL;
    ICatalogCollection* pCatalogCollectionCo
= NULL;
    ICatalogCollection* pCatalogCollectionItf
= NULL;
    ICatalogCollection*
pCatalogCollectionMethod = NULL;

    ICatalogObject*
pCatalogObjectApp = NULL;
    ICatalogObject*
pCatalogObjectCo = NULL;
    ICatalogObject*
pCatalogObjectItf = NULL;
    ICatalogObject*
pCatalogObjectMethod = NULL;

    _bstr_t
bstrTemp, bstrTemp2, bstrTemp3, bstrTemp4;
    _bstr_t
bstrDllPath = szDllPath;
    _variant_t
vTmp, vKey;
    long
lActProp, lCount, lCountCo, lCountItf,
lCountMethod;
```

```
bool
bTmp;

    CoInitializeEx(NULL, COINIT_MULTITHREADED);

    HRESULT hr =
CoCreateInstance(CLSID_COMAdminCatalog,
                NULL,
                CLSCTX_INPROC_SERVER,
                IID_ICOMAdminCatalog,
                (void**)
                &pCOMAdminCat);

    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "Applications";

    // Attempt to connect to "Applications" in
the Catalog
    hr = pCOMAdminCat->GetCollection(bstrTemp,
                (IDispatch**)
                &pCatalogCollectionApp);
    if (!SUCCEEDED(hr)) goto Error;

    // Attempt to load the "Applications"
collection
    hr = pCatalogCollectionApp->Populate();
    if (!SUCCEEDED(hr)) goto Error;

    hr = pCatalogCollectionApp-
>get_Count(&lCount);
    if (!SUCCEEDED(hr)) goto Error;

    // iterate through applications to delete
existing "TPC-C" application (if any)
    while (lCount > 0)
    {
        hr = pCatalogCollectionApp-
>get_Item(lCount - 1, (IDispatch**)
                &pCatalogObjectApp);
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogObjectApp-
>get_Name(&vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        if (wcsncmp(vTmp.bstrVal, L"TPC-
C"))
        {
            lCount--;
            continue;
        }
        else
        {
```

```
                hr =
pCatalogCollectionApp->Remove(lCount - 1);
                if (!SUCCEEDED(hr))
                break;
            }
        }

        hr = pCatalogCollectionApp-
>SaveChanges(&lActProp);
        if (!SUCCEEDED(hr)) goto Error;

        // add the new application
        hr = pCatalogCollectionApp-
>Add((IDispatch**) &pCatalogObjectApp);
        if (!SUCCEEDED(hr)) goto Error;

        // set properties
        bstrTemp = "Name";
        vTmp = "TPC-C";
        hr = pCatalogObjectApp->put_Value(bstrTemp,
                vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        // set as a library (in process)
        application
        bstrTemp = "Activation";
        lActProp = COMAdminActivationInproc;
        vTmp = lActProp;
        hr = pCatalogObjectApp->put_Value(bstrTemp,
                vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        // set security level to process
        bstrTemp = "AccessChecksLevel";
        lActProp =
COMAdminAccessChecksApplicationLevel;
        vTmp = lActProp;
        hr = pCatalogObjectApp->put_Value(bstrTemp,
                vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        // save key to get the Components
        collection later
        hr = pCatalogObjectApp->get_Key(&vKey);
        if (!SUCCEEDED(hr)) goto Error;

        // save changes (app creation) so component
        installation will work
        hr = pCatalogCollectionApp-
>SaveChanges(&lActProp);
        if (!SUCCEEDED(hr)) goto Error;

        pCatalogObjectApp->Release();
        pCatalogObjectApp = NULL;

        bstrTemp = "TPC-C";
        // app name
        bstrTemp2 = bstrDllPath +
"tpcc_com_all.dll"; // DLL
        bstrTemp3 = bstrDllPath +
"tpcc_com_all.tlb"; // type library (TLB)
```

```

        bstrTemp4 =      bstrDllPath +
"tpcc_com_ps.dll";    // proxy/stub dll

        hr = pCOMAdminCat-
>InstallComponent(bstrTemp,

        bstrTemp2,

        bstrTemp3,

        bstrTemp4);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "Components";
        hr = pCatalogCollectionApp-
>GetCollection(bstrTemp, vKey, (IDispatch**)
&pCatalogCollectionCo);
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionCo->Populate();
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionCo-
>get_Count(&lCountCo);
        if (!SUCCEEDED(hr)) goto Error;

        // iterate through components in
application and set the properties
        while (lCountCo > 0)
        {
            hr = pCatalogCollectionCo-
>get_Item(lCountCo - 1, (IDispatch**)
&pCatalogObjectCo);
            if (!SUCCEEDED(hr)) goto Error;

            // used for debugging (view the
name)
            hr = pCatalogObjectCo-
>get_Name(&vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "ConstructionEnabled";
            bTmp = TRUE;
            vTmp = bTmp;
            hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp = "ConstructorString";
            bstrTemp2 = "dummy string (do not
remove)";

            vTmp = bstrTemp2;
            hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
            if (!SUCCEEDED(hr)) goto Error;

            bstrTemp =
"JustInTimeActivation";
            bTmp = TRUE;

```

```

        vTmp = bTmp;
        hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp = "MaxPoolSize";
        vTmp.Clear(); // clear
variant so it isn't stored as a bool (_variant_t
feature)

        vTmp = (long)30;
        hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp =
"ObjectPoolingEnabled";
        bTmp = TRUE;
        vTmp = bTmp;
        hr = pCatalogObjectCo-
>put_Value(bstrTemp, vTmp);
        if (!SUCCEEDED(hr)) goto Error;

        // save key to get the
InterfacesForComponent collection
        hr = pCatalogObjectCo-
>get_Key(&vKey);
        if (!SUCCEEDED(hr)) goto Error;

        bstrTemp =
"InterfacesForComponent";
        hr = pCatalogCollectionCo-
>GetCollection(bstrTemp, vKey, (IDispatch**)
&pCatalogCollectionItf);
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionItf-
>Populate();
        if (!SUCCEEDED(hr)) goto Error;

        hr = pCatalogCollectionItf-
>get_Count(&lCountItf);
        if (!SUCCEEDED(hr)) goto Error;

        // iterate through interfaces in
component
        while (lCountItf > 0)
        {
            hr =
pCatalogCollectionItf->get_Item(lCountItf - 1,
(IDispatch**) &pCatalogObjectItf);
            if (!SUCCEEDED(hr))
                goto Error;

            // save key to get the
MethodsForInterface collection
            hr = pCatalogObjectItf-
>get_Key(&vKey);
            if (!SUCCEEDED(hr))
                goto Error;

```

```

        bstrTemp =
"MethodsForInterface";
        hr =
pCatalogCollectionItf->GetCollection(bstrTemp, vKey,
(IDispatch**) &pCatalogCollectionMethod);
        if (!SUCCEEDED(hr))
            goto Error;

        hr =
pCatalogCollectionMethod->Populate();
        if (!SUCCEEDED(hr))
            goto Error;

        hr =
pCatalogCollectionMethod->get_Count(&lCountMethod);
        if (!SUCCEEDED(hr))
            goto Error;

        // iterate through
methods of interface
        while (lCountMethod >
0)
        {
            hr =
pCatalogCollectionMethod->get_Item(lCountMethod - 1,
(IDispatch**) &pCatalogObjectMethod);
            if
                (!SUCCEEDED(hr)) goto Error;

            bstrTemp =
"AutoComplete";
            bTmp = TRUE;
            vTmp = bTmp;
            hr =
pCatalogObjectMethod->put_Value(bstrTemp, vTmp);
            if
                (!SUCCEEDED(hr)) goto Error;

            pCatalogObjectMethod->Release();
            pCatalogObjectMethod = NULL;

            lCountMethod--
        }

        // save changes
        hr =
pCatalogCollectionMethod->SaveChanges(&lActProp);
        if (!SUCCEEDED(hr))
            goto Error;

        pCatalogObjectItf-
>Release();
        pCatalogObjectItf =
NULL;

        lCountItf--;
    }

```

```

        pCatalogObjectCo->Release();
        pCatalogObjectCo = NULL;

        lCountCo--;
    }

    // save changes
    hr = pCatalogCollectionCo-
>SaveChanges(&lActProp);
    if (!SUCCEEDED(hr)) goto Error;

    pCatalogCollectionApp->Release();
    pCatalogCollectionApp = NULL;

    pCatalogCollectionCo->Release();
    pCatalogCollectionCo = NULL;

    pCatalogCollectionItf->Release();
    pCatalogCollectionItf = NULL;

    pCatalogCollectionMethod->Release();
    pCatalogCollectionMethod = NULL;

Error:
    CoUninitialize();

    if (!SUCCEEDED(hr))
    {
        LPTSTR lpBuf;
        DWORD dwRes =
FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM,
                NULL,
                hr,
                MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPTSTR)
&lpBuf,
                0,
                NULL);
        //      _tprintf(_T("Error adding
components. HRESULT: 0x%x\n%s"), hr, lpBuf);
        return TRUE;
    }
    else
        return FALSE;
}

```

license.txt

END-USER LICENSE AGREEMENT FOR
MICROSOFT TPC-C BENCHMARK KIT

IMPORTANT READ CAREFULLY: This Microsoft End-

User License Agreement (EULA) is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the Microsoft software product identified above, which includes computer software and may include associated media, printed materials, and online or electronic documentation (SOFTWARE PRODUCT). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this Agreement, you are not authorized to use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1. GRANT OF LICENSE. This EULA grants you the following rights:
Use. Microsoft grants to you the right to install and use copies of the SOFTWARE PRODUCT only in conjunction with validly licensed copies of Microsoft SQL Server and/or Microsoft Windows NT Server software. You may also make copies of the SOFTWARE PRODUCT for backup and archival purposes.

2. RESTRICTIONS.
--You must maintain all copyright notices on all copies of the SOFTWARE PRODUCT.
--You may not distribute copies of the SOFTWARE PRODUCT to third parties.
--You may not rent, lease or lend the SOFTWARE PRODUCT.
--You may not use the SOFTWARE PRODUCT or any derivative works thereof to internally test database management system software other than Microsoft SQL Server and/or operating system software other than Microsoft Windows NT.
-- You may not disclose the results of any benchmark tests using the SOFTWARE PRODUCT to any third party without Microsoft's prior written approval.
-- You may not disclose or provide the SOFTWARE PRODUCT or any derivative works thereof, or any information relating to the SOFTWARE PRODUCT (including the existence of the SOFTWARE PRODUCT or the results of use and testing or benchmark testing), to any third party without Microsoft's written permission.

3. TERMINATION. Without prejudice to any other rights,

Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT.

4. COPYRIGHT. All title and copyrights in and to the SOFTWARE PRODUCT and any copies thereof are owned by Microsoft or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content.

5. UPGRADES. If the SOFTWARE PRODUCT is labeled as an upgrade, you must be properly licensed to use a product identified by Microsoft as being eligible for the upgrade in order to use the SOFTWARE PRODUCT. A SOFTWARE PRODUCT labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this EULA.

6. U.S. GOVERNMENT RESTRICTED RIGHTS. The SOFTWARE PRODUCT is provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Microsoft Corporation/One Microsoft Way/Redmond, WA 98052-6399.

7. EXPORT RESTRICTIONS. You agree that you will not export or re-export the SOFTWARE PRODUCT to any country, person, entity or end user subject to U.S.A. export restrictions. Restricted countries currently include, but are not necessarily limited to Cuba, Iran, Iraq, Libya, North Korea, Syria, and the Federal Republic of Yugoslavia (Serbia and Montenegro, U.N. Protected Areas and areas of Republic of Bosnia and Herzegovina under the

control of Bosnian Serb forces). You warrant and represent that neither the U.S.A. Bureau of Export Administration nor any other federal agency has suspended, revoked or denied your export privileges.

8. NO WARRANTY. ANY USE OF THE SOFTWARE PRODUCT IS AT YOUR OWN RISK. THE SOFTWARE PRODUCT IS PROVIDED FOR USE ONLY WITH MICROSOFT SQL SERVER AND/OR MICROSOFT WINDOWS NT SERVER SOFTWARE. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, MICROSOFT AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.

9. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE PRODUCT, EVEN IF MICROSOFT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

10. LIMITATION OF LIABILITY. MICROSOFT'S ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY UNDER THIS EULA SHALL NOT EXCEED FIVE DOLLARS (US\$5.00).

11. MISCELLANEOUS
This EULA is governed by the laws of the State of Washington, U.S.A.
Should you have any questions concerning this EULA, or if you desire to contact Microsoft for any reason, please contact the Microsoft subsidiary serving your country, or write:
Microsoft Sales Information Center/One Microsoft Way/Redmond, WA 98052-6399.

Si vous avez acquis votre produit Microsoft au CANADA, la garantie limitée suivante vous concerne:

EXCLUSION DE GARANTIES. Microsoft renonce entièrement ... toute garantie pour le LOGICIEL. Le

LOGICIEL et toute autre documentation s'y rapportant sont fournis @ comme tels - sans aucune garantie quelle qu'elle soit, expresse ou implicite, y compris, mais ne se limitant pas aux garanties implicites de la qualité, marchande ou un usage particulier. Le risque total d'écoulement de l'utilisation ou de la performance du LOGICIEL est entre vos mains.

RESPONSABILITÉ LIMITÉE. La seule obligation de Microsoft et votre recours exclusif concernant ce contrat n'excéderont pas cinq dollars (US\$5.00).

ABSENCE DE RESPONSABILITÉ POUR LES DOMMAGES INDIRECTS. Microsoft ou ses fournisseurs ne pourront être tenus responsables en aucune circonstance de tout dommage quel qu'il soit (y compris mais non de façon limitative les dommages directs ou indirects causés par la perte de biens, fices commerciaux, l'interruption des affaires, la perte d'information commerciale ou toute autre perte pécuniaire) résultant de l'utilisation ou de l'impossibilité d'utilisation de ce produit, et ce, même si la société, Microsoft a, à l'époque de l'achat, de tels dommages. Certains États/jurisdictions ne permettent pas l'exclusion ou la limitation de responsabilité relative aux dommages indirects ou consécutifs, et la limitation ci-dessus peut ne pas s'appliquer ... votre pays. La présente Convention est régie par les lois de la province d'Ontario, Canada.

Chacune des parties ... la présente reconnaît irrévocablement la compétence des tribunaux de la province d'Ontario et consent ... instituer tout litige qui pourrait découler de la présente auprès des tribunaux situés dans le district judiciaire de York, province d'Ontario. Au cas où vous auriez des questions concernant cette licence ou que vous désiriez vous mettre en rapport avec Microsoft pour quelque raison que ce soit, veuillez contacter la succursale Microsoft desservant votre pays, dont l'adresse est fournie dans ce produit, ou écrire ...

Microsoft Customer Sales and Service, One Microsoft Way, Redmond, Washington 98052 6399.

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,
    ERR_MEM_ALLOC_FAILED
};

class CCOMPONENT_ERR : public CBaseErr
{
public:
    CCOMPONENT_ERR(COMPONENT_ERROR
Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
        m_SystemErr = 0;
        m_szErrorText = NULL;
    };

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

    ~CCOMPONENT_ERR()
    {
        if (m_szTextDetail !=
NULL)

```

```

delete []
m_szTextDetail;
        if (m_szErrorText !=
NULL)
delete []
m_szErrorText;
        };
        COMPONENT_ERROR m_Error;
        char
*m_szTextDetail;
        char
*m_szErrorText;
        DWORD
m_SystemErr;

        int ErrorType() {return
ERR_TYPE_COMPONENT};
        char *ErrorTypeStr() { return
"COMPONENT"; }
        int ErrorNum() {return m_Error;};
        char *ErrorText();

};

static void WriteMessageToEventLog(LPTSTR lpszMsg);

////////////////////////////////////
////////////////////////////////////
// CTPCC_Common
class CTPCC_Common :
public ITPCC,
public IObjectControl,
public IObjectConstruct,
public
CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
COM_INTERFACE_ENTRY(ITPCC)
COM_INTERFACE_ENTRY(IObjectControl)
COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

        CTPCC_Common();
        ~CTPCC_Common();

// ITPCC
public:
        HRESULT __stdcall NewOrder(
VARIANT txn_in, VARIANT* txn_out);
        HRESULT __stdcall Payment(
VARIANT txn_in, VARIANT* txn_out);
        HRESULT __stdcall Delivery(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out);
        HRESULT __stdcall OrderStatus(
VARIANT txn_in, VARIANT* txn_out);

        HRESULT __stdcall CallSetComplete();

```

```

// IObjectControl
        STDMETHODCALLTYPE CanBePooled() { return
m_bCanBePooled; }
        STDMETHODCALLTYPE Activate() { return S_OK; }
        // we don't support COM Services
transactions (no enlistment)
        STDMETHODCALLTYPE 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;
        };

};

////////////////////////////////////
////////////////////////////////////
// CTPCC
class CTPCC :
public CTPCC_Common,
public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

        BEGIN_COM_MAP(CTPCC)
//COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx<CComSingleThreadModel>)
COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};

////////////////////////////////////
////////////////////////////////////
// CNewOrder
class CNewOrder :

```

```

        public CTPCC_Common,
        public CComCoClass<CNewOrder,
&CLSID_NewOrder>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

        BEGIN_COM_MAP(CNewOrder)
// COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
// HRESULT __stdcall NewOrder(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall Payment(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
        HRESULT __stdcall OrderStatus(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

////////////////////////////////////
////////////////////////////////////
// COrderStatus
class COrderStatus :
public CTPCC_Common,
public CComCoClass<COrderStatus,
&CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

        BEGIN_COM_MAP(COrderStatus)
// COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
        HRESULT __stdcall NewOrder(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall Payment(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
        HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
// HRESULT __stdcall OrderStatus(
VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

```

```

////////////////////////////////////
////////////////////////////////////
// CPayment
class CPayment :
    public CTPCC_Common,
    public CComCoClass<CPayment,
    &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
// COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    // HRESULT __stdcall Payment(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

////////////////////////////////////
////////////////////////////////////
// CStockLevel
class CStockLevel :
    public CTPCC_Common,
    public CComCoClass<CStockLevel,
    &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
// COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    // HRESULT __stdcall StockLevel( VARIANT
txn_in, VARIANT* txn_out) {return E_NOTIMPL;}
}

```

```

        HRESULT __stdcall OrderStatus(
        VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
};

```

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 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 TPC 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;
always has to be ODBC
    pReg->eDB_Protocol = ODBC;
    size = sizeof(szTmp);
    //if ( RegQueryValueEx(hKey, "DB_Protocol",
0, &type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
    //{
        //if ( !strcmp(szTmp,
szDBNames[ODBC]) )

```

```

        // pReg->eDB_Protocol =
ODBC;
    //}

    pReg->eTxnMon = None;
    // determine txn monitor to use; may be
either COM, or blank
    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "TxnMonitor", 0,
&type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
    {
        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, "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;
}

```

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

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

//This structure defines the data necessary to keep
distinct for each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
        enum DBPROTOCOL eDB_Protocol;
        enum TXNMNOM eTxnMon;
        BOOL bCOM_SinglePool;
        DWORD dwMaxConnections;
        DWORD dwMaxPendingDeliveries;
        DWORD dwNumberOfDeliveryThreads;
        char szPath[128];
        char szDbServer[32];
        char szDbName[32];
        char szDbUser[32];
        char szDbPassword[32];
        wchar_t szSPPrefix[32];
        //tpcc_odbc.dll stored procedures prefix
        DWORD dwConnectDelay; // delay in
ms to use in pacing connection open and close
        BOOL bCallNoDuplicatesNewOrder; //
whether to check for non-duplicate item ids and call
a different New Order SP
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg
);

```

RESOURCE.H

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by install.rc
//
#define IDD_DIALOG1 101
#define IDI_ICON1 102
#define IDR_TPCCDLL 103
#define IDD_DIALOG2 105
#define IDI_ICON2 106
#define IDR_DELIVERY 107
#define IDD_DIALOG3 108
#define IDR_LICENSES1 112
#define IDD_DIALOG4 113
#define IDR_TPCCOBJ1 117
#define IDR_TPCCSTUB1 118
#define IDR_ODBC_DLL 123
#define IDR_COM_DLL 126
#define IDR_COMPS_DLL 127
#define IDR_COMALL_DLL 128
#define IDR_COMTYPLIB_DLL 129
#define IDR_MSVC71 130
#define ED_BN_LOG 1001
#define ED_KEEP 1002
#define ED_THREADS 1003

```

```

#define ED_THREADS2 1004
#define IDC_PATH 1007
#define IDC_VERSION 1009
#define IDC_RESULTS 1010
#define IDC_PROGRESS1 1011
#define IDC_STATUS 1012
#define IDC_BUTTON1 1013
#define ED_MAXCONNECTION 1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES 1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT 1018
#define ED_IIS_LISTEN_BACKLOG 1019
#define IDC_DBLIB 1021
#define IDC_LICENSE 1022
#define IDC_ODBC 1022
#define IDC_CONNECT_POOL 1023
#define ED_DB_SERVER 1023
#define ED_USER_CONNECT_DELAY_TIME 1024
#define ED_DB_USER_ID 1024
#define IDC_MTS 1025
#define IDC_TM_MTS 1025
#define IDC_TM_TUXEDO 1026
#define IDC_TM_NONE 1027
#define ED_DB_PASSWORD 1028
#define ED_DB_NAME 1029
#define IDC_TM_ENCINA 1030

```

// Next default values for new objects

```

//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 131
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1031
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

tpcc.cpp

```

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

```

```

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

#include <sqltypes.h>

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

#include "..\..\common\src\trans.h"
//tpckit transaction header contains
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_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 "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_ODBC *pCTPCC_ODBC_new;
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();

        if
(Reg.eTxnMon == COM)
        {
            strcpy( szDllName, Reg.szPath );
            strcat( szDllName, "tpcc_com.dll");
            hLibInstanceTm = LoadLibrary( szDllName );
            if
(hLibInstanceTm == NULL)
                throw new CWBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

            //
            get function pointer to wrapper for class constructor
            pCTPCC_COM_new = (TYPE_CTPCC_COM*)
GetProcAddress(hLibInstanceTm, "CTPCC_COM_new");
            if
(pCTPCC_COM_new == NULL)
                throw new CWBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );

            // load DLL
            for database connection
            if
((Reg.eTxnMon == None) || (dwNumDeliveryThreads > 0))
            {
                if
(Reg.eDB_Protocol == ODBC)
                {
                    strcpy( szDllName, Reg.szPath );
                    strcat( szDllName, "tpcc_odbc.dll");
                    hLibInstanceDb = LoadLibrary( szDllName );
                    if (hLibInstanceDb == NULL)
                        throw new CWBCLNT_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 CWBCLNT_ERR(
ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
        }
    }

    // Check
    whether Service Pack 1 has been installed if
    Windows Server 2003. The RTM version has
    // running on
    // a
    limitation on concurrent HTTP connections.
    //
    OSVERSIONINFOEX VersionInfo;

    VersionInfo.dwOSVersionInfoSize =
sizeof(OSVERSIONINFOEX);
    if
(GetVersionEx((LPOSVERSIONINFO)&VersionInfo))
    {
        if
        (VersionInfo.dwMajorVersion == 5 && // Windows
        2000/2003 Server?
        VersionInfo.dwMinorVersion == 2 && //
        Windows 2003 Server?
        VersionInfo.wServicePackMajor == 0) //
        Service Pack installed?
        {
            TCHAR szMsg[256];

            _sntprintf(szMsg, sizeof(szMsg),
                "\nRunning on
                Windows Server 2003 without at least Service Pack
                1\n"
                "limits the
                number of concurrent HTTP connections to around
                8000");

            // Use event logging to log the error.
            //
            HANDLE hEventSource =
RegisterEventSource(NULL, TEXT("TPCC.DLL"));
            LPTSTR lpszStrings[1] = { szMsg };

            if (hEventSource != NULL)

```

```

        {
            ReportEvent(hEventSource, //
handle of event source
            EVENTLOG_WARNING_TYPE,
            // event type
            0,
            // event category
            0,
            // event ID
            NULL,
            // current user's SID
            1,
            // strings in lpszStrings
            0,
            // no bytes of raw data
            (LPCTSTR *)lpszStrings,
            // array of error strings
            NULL);
            // no raw data
            (VOID)
DeregisterEventSource(hEventSource);
        }
    }

    if
(dwNumDeliveryThreads)
    {
        Initialize delivery delay critical section
        //
        InitializeCriticalSection(&hConnectCritical
Section);
        //
        for deferred delivery txns:
        //
        hDoneEvent = CreateEvent( NULL, TRUE /*
manual reset */, FALSE /* initially not signalled */,
NULL );
        InitializeCriticalSection(&DelBuffCriticalS
ection);
        hWorkerSemaphore = CreateSemaphore( NULL,
0, dwDelBuffSize, NULL );
    }

```

```

        dwDelBuffFreeCount = dwDelBuffSize;

        InitJulianTime(NULL);

        //
        create unique log file name based on delilog-yyymmdd-
        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(&DelBuffCriticalSecti
                on);

                //
                Delete delivery delay critical section
                //
                DeleteCriticalSection(&hConnectCriticalSecti
                on);

                DeleteCriticalSection(&TermCriticalSection)
                ;

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

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

```

```

        Sleep(500);
        break;

        default: /* nothing
        */;
    }
}
catch (CBaseErr *e)
{
    TCHAR szMsg[256];
    _sntprintf(szMsg, sizeof(szMsg),
    "%s error, code %d: %s",
    e-
    >ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
    WriteMessageToEventLog( szMsg );
    delete e;
    TerminateExtension(0);
    return FALSE;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
    exception. DLL could not load.));
    TerminateExtension(0);
    return FALSE;
}

return TRUE;

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

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

    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 TermId,
iSyncId;
    char szBuffer[4096];

    int lpbSize;
    static char szHeader[] = "200 Ok";
    DWORD dwSize = 6;
    // initial value is strlen(szHeader)
    char szHeader1[4096];
    DWORD dwAddr; // used to
store Win32 exception address

```

```

LPEXCEPTION_POINTERS
pExceptionInfo; // pointer to Win32
exception info

#ifdef ICECAP
StartCAP();
#endif

// Use structured exception handling for
Win32 exceptions
//
try
{
    ProcessCommand(pECB, szBuffer,
TermId, iSyncId);
}
except ( pExceptionInfo =
GetExceptionInformation(), // can call
GetExceptionInformation only in filter (not handler)
dwAddr =
(DWORD)pExceptionInfo->ExceptionRecord-
>ExceptionAddress, // save the address

EXCEPTION_EXECUTE_HANDLER) // handle all
exceptions
{
    char
szMsg[512];
    int
iLen;

    MEMORY_BASIC_INFORMATION mbi ;
    VirtualQuery( (void*)dwAddr,
&mbi, sizeof( mbi ) );
    DWORD hInstance =
(DWORD)mbi.AllocationBase ;

    iLen = wsprintf(szMsg,
TEXT("Unhandled exception (%#x) in Web Client's
HttpExtensionProc. "
"Occured at
address %#x, base %#x, tpcc_com.dll at %#x, tpcc.dll
at %#x, tpcc_com_all.dll at %#x"),
GetExceptionCode(), dwAddr, hInstance,
GetModuleHandle("tpcc_com.dll"),
GetModuleHandle("tpcc.dll"),
GetModuleHandle("tpcc_com_all.dll"));

    if (txxDelilog != NULL)
    {
        txxDelilog-
>WriteCtrlRecToLog(TXN_EVENT_WARNING, szMsg, iLen +
1);
    }
    ErrorForm( pECB, ERR_TYPE_WEBDLL,
GetExceptionCode(), TermId, iSyncId, szMsg, szBuffer
);
}

```

```

#ifdef ICECAP
StopCAP();
#endif

lpbSize = strlen(szBuffer);
dwSize += lpbSize;
dwSize += 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;
}

/* FUNCTION: ProcessCommand
*
* PURPOSE:                       This function parses the commands
from the driver and executes corresponding
transactions.
*
* ARGUMENTS:                     EXTENSION_CONTROL_BLOCK
*pECB structure pointer to passed in
internet
*
*                               service information.
*
* RETURNS:                       None (outputs into the
szBuffer parameter).
*
* COMMENTS:                       Separated from HttpExtensionProc
to be able to use structured exception handling in
*
HttpExtensionProc (cannot mix C++ and Win32
exceptions in one functions).
*/
void ProcessCommand(EXTENSION_CONTROL_BLOCK *pECB,
char* szBuffer, int& TermId, int& iSyncId)
{
    int iCmd, FormId;

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

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

```

```

        {
            //
            debugging...
            char
            szTmp[128];
            wsprintf(
            szTmp, "Invalid term ID; TermId = %d", TermId );
            WriteMessageToEventLog( szTmp );
            throw new
            CWEBCLNT_ERR( ERR_INVALID_TERMID );
        }

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

        //set use time
        Term.pClientData[TermId].iTickCount =
        GetTickCount();
    }

    switch(iCmd)
    {
    case 0:
        WelcomeForm(pECB,
        szBuffer);
        break;
    case 1:
        switch( FormId )
        {
        case WELCOME_FORM:
        case MAIN_MENU_FORM:
            break;
        case NEW_ORDER_FORM:
            ProcessNewOrderForm(pECB, TermId,
            szBuffer);
            break;
        case PAYMENT_FORM:
            ProcessPaymentForm(pECB, TermId, szBuffer);
            break;
        case DELIVERY_FORM:
            ProcessDeliveryForm(pECB, TermId,
            szBuffer);
            break;
        case ORDER_STATUS_FORM:
            ProcessOrderStatusForm(pECB, TermId,
            szBuffer);
            break;
        case STOCK_LEVEL_FORM:
            ProcessStockLevelForm(pECB, TermId,
            szBuffer);
        }
    }
}

```

```

        break;
    }
    break;
    case 2: // new-order selected
        from menu; display new-order input form
        MakeNewOrderForm(TermId, NULL, INPUT_FORM,
        szBuffer);
        break;
    case 3: // payment selected
        from menu; display payment input form
        MakePaymentForm(TermId,
        NULL, INPUT_FORM, szBuffer);
        break;
    case 4: // delivery selected
        from menu; display delivery input form
        MakeDeliveryForm(TermId, NULL, INPUT_FORM,
        szBuffer);
        break;
    case 5: // order-status
        selected from menu; display order-status input form
        MakeOrderStatusForm(TermId, NULL,
        INPUT_FORM, szBuffer);
        break;
    case 6: // stock-level selected
        from menu; display stock-level input form
        MakeStockLevelForm(TermId, NULL,
        INPUT_FORM, szBuffer);
        break;
    case 7: // ExitCmd
        TermDelete(TermId);
        WelcomeForm(pECB,
        szBuffer);
        break;
    case 8:
        SubmitCmd(pECB,
        szBuffer);
        break;
    case 9: // menu
        MakeMainMenuForm(TermId,
        Term.pClientData[TermId].iSyncId, szBuffer);
        break;
    case 10: // CMD=Clear
        // resets all
        connections; should only be used when no other
        connections are active
        TermDeleteAll();
        TermInit();
        WelcomeForm(pECB,
        szBuffer);
        break;
}

```

```

        case 11: // CMD=Stats
            StatsCmd(pECB,
            szBuffer);
            break;
        }
    }
    catch (CBaseErr *e)
    {
        ErrorForm( pECB, e->ErrorType(),
        e->ErrorNum(), TermId, iSyncId, e->ErrorText(),
        szBuffer );
        delete e;
    }
}

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

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

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

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}

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

```

```

*           the delivery txn,
information is logged to record the txn status and
execution
*           time.
*/

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

    DELIVERY_TRANSACTION
    delivery;
    PDELIVERY_DATA
    pDeliveryData;
    TXN_RECORD_TPCC_DELIV_DEF    txnDeliRec;

    DWORD
    index;
    HANDLE
    handles[2];

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

    assert(txnDeliRec != NULL);

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

                Sleep(Reg.dwConnectDelay);

                LeaveCriticalSection(&hConnectCriticalSection);
            }

            pTxn = pCTPCC_ODBC_new(
                Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
                szMyComputerName, Reg.szDbName,
                Reg.szSPPrefix,
                Reg.bCallNoDuplicatesNewOrder );

            pDeliveryData = pTxn-
            >BuffAddr_Delivery();
        }
    }

```

```

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

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

                    ZeroMemory(&txnDeliRec,
                    sizeof(txnDeliRec));

                    txnDeliRec.TxnType =
                    TXN_REC_TYPE_TPCC_DELIV_DEF;

                    // make a
                    local copy of current entry from delivery buffer and
                    increment buffer index
                    EnterCriticalSection(&DelBuffCriticalSection);

                    delivery =
                    *(pDelBuff+dwDelBuffBusyIndex);

                    dwDelBuffFreeCount++;
                }
            }
        }

```

```

            dwDelBuffBusyIndex++;
            if
            (dwDelBuffBusyIndex == dwDelBuffSize) // wrap-
            around if at end of buffer
            dwDelBuffBusyIndex = 0;

            LeaveCriticalSection(&DelBuffCriticalSection);
        n);

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

```

```

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

        // log the error txn
        txnDeliRec.TxnStatus =
e->ErrorType();
        if (txnDeliLog != NULL)
            txnDeliLog->WriteToLog(&txnDeliRec);
        delete e;
    }
    catch (...)
    {
        // unhandled exception;
        shouldn't happen; not much we can do...
        WriteMessageToEventLog(TEXT("Unhandled
exception caught in DeliveryWorkerThread."));
    }
}

ErrorExit:
if (Reg.dwConnectDelay > 0)
{
    // Synchronize disconnect (for
VIA)
    //
    EnterCriticalSection(&hConnectCriticalSection);
    Sleep(Reg.dwConnectDelay);
}
delete pTxn;
if (Reg.dwConnectDelay > 0)
{
    // Synchronize disconnect (for
VIA)
    //
    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;
        (pDelBuff+dwDelBuffFreeIndex)-
        = 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\""
VALUE=\"0\""

        "<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 == 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 );
}
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 iTotals;

    EnterCriticalSection(&TermCriticalSection);

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

            iTotals++;

    }

    LeaveCriticalSection(&TermCriticalSection);

    wsprintf( szBuffer,

"<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"

" <BODY><B><BIG> Total
Active Connections: %d </BIG></B><BR></BODY></HTML>"
, iTotals );
}

char *CWEBCLNT_ERR::ErrorText()
{
    static SERRORMSG errorMsgs[] =
{
        { ERR_COMMAND_UNDEFINED,
"Command undefined."
},
        { ERR_D_ID_INVALID,
"Invalid District ID Must be 1 to 10."
},
        {
ERR_DELIVERY_CARRIER_ID_RANGE,
"Delivery Carrier ID out of range
must be 1 - 10."
},
        {
ERR_DELIVERY_CARRIER_INVALID,
"Delivery Carrier ID invalid must be
numeric 1 - 10."
},
        {
ERR_DELIVERY_MISSING_OCD_KEY,
"Delivery missing Carrier ID key \"OCD*\"."
},
        {
ERR_DELIVERY_THREAD_FAILED,
"Could not start delivery worker
thread."
},
        { ERR_GETPROCADDR_FAILED,

```

```

"Could not map proc in DLL. GetProcAddr
error. DLL="
{ ERR_HTML_ILL_FORMED,
},
string."Required key field is missing from HTML
",
{
ERR_INVALID_SYNC_CONNECTION,
"Invalid Terminal Sync ID."
},
{
ERR_INVALID_TERMINAL_ID,
"Invalid Terminal ID."
},
{
ERR_LOADDLL_FAILED,
"Load of DLL failed. DLL="
},
{
ERR_MAX_CONNECTIONS_EXCEEDED,
"No connections available. Max Connections
is probably too low."
},
{
ERR_MISSING_REGISTRY_ENTRIES,
"Required registry entries are missing.
Rerun INSTALL to correct."
},
{
ERR_NEWORDER_CUSTOMER_INVALID,
"New Order Customer id invalid
data type, range = 1 to 3000."
},
{
ERR_NEWORDER_CUSTOMER_KEY,
"New Order missing Customer key
\"CID*\"."
},
{
ERR_NEWORDER_DISTRICT_INVALID,
"New Order District ID Invalid
range 1 - 10."
},
{
ERR_NEWORDER_FORM_MISSING_DID,
"New Order missing District key
\"DID*\"."
},
{
ERR_NEWORDER_ITEMID_INVALID,
"New Order Item Id is wrong data type, must
be numeric."
},
{
ERR_NEWORDER_ITEMID_RANGE,
"New Order Item Id is out of
range. Range = 1 to 999999."
},
{
ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
"New Order Item Id field entered without a
corresponding Supp_W."
},
{
ERR_NEWORDER_MISSING_IID_KEY,
"New Order missing Item Id key \"IID*\"."
}

```

```

    },
    {
ERR_NEWORDER_MISSING_QTY_KEY,
    "New Order Missing Qty key \"Qty##*\"."
    },
    {
ERR_NEWORDER_MISSING_SUPPW_KEY,
    "New Order missing Supp_W key
\"SP##*\"."
    },
    {
ERR_NEWORDER_NOITEMS_ENTERED,
    "New Order No order lines entered."
    },
    {
ERR_NEWORDER_QTY_INVALID,
    "New Order Qty invalid must be
numeric range 1 - 99."
    },
    {
ERR_NEWORDER_QTY_RANGE,
    "New Order Qty is out of range. Range = 1
to 99."
    },
    {
ERR_NEWORDER_QTY_WITHOUT_SUPPW,
    "New Order Qty field entered
without a corresponding Supp_W."
    },
    {
ERR_NEWORDER_SUPPW_INVALID,
    "New Order Supp_W invalid data
type must be numeric."
    },
    {
ERR_NO_SERVER_SPECIFIED,
    "No Server name specified."
    },
    {
ERR_ORDERSTATUS_CID_AND_CLT,
    "Order Status Only Customer ID or Last Name
may be entered, not both."
    },
    {
ERR_ORDERSTATUS_CID_INVALID,
    "Order Status Customer ID invalid, range
must be numeric 1 - 3000."
    },
    {
ERR_ORDERSTATUS_CLT_RANGE,
    "Order Status Customer last name
longer than 16 characters."
    },
    {
ERR_ORDERSTATUS_DID_INVALID,
    "Order Status District invalid, value must
be numeric 1 - 10."
    },
    {
ERR_ORDERSTATUS_MISSING_CID_CLT,
    "Order Status Either Customer ID or Last
Name must be entered."
    },
    {
ERR_ORDERSTATUS_MISSING_CID_KEY,
    "Order Status missing Customer key

```

```

\"CID*\"."
    },
    {
ERR_ORDERSTATUS_MISSING_CLT_KEY,
    "Order Status missing Customer Last Name
key \"CLT*\"."
    },
    {
ERR_ORDERSTATUS_MISSING_DID_KEY,
    "Order Status missing District key
\"DID*\"."
    },
    {
ERR_PAYMENT_CDI_INVALID,
    "Payment Customer district
invalid must be numeric."
    },
    {
ERR_PAYMENT_CID_AND_CLT,
    "Payment Only Customer ID or Last
Name may be entered, not both."
    },
    {
ERR_PAYMENT_CUSTOMER_INVALID,
    "Payment Customer data type invalid, must
be numeric."
    },
    {
ERR_PAYMENT_CWI_INVALID,
    "Payment Customer Warehouse
invalid, must be numeric."
    },
    {
ERR_PAYMENT_DISTRICT_INVALID,
    "Payment District ID is invalid, must be 1
- 10."
    },
    {
ERR_PAYMENT_HAM_INVALID,
    "Payment Amount invalid data type
must be numeric."
    },
    {
ERR_PAYMENT_HAM_RANGE,
    "Payment Amount out of range, 0 - 9999.99."
    },
    {
ERR_PAYMENT_LAST_NAME_TO_LONG,
    "Payment Customer last name
longer than 16 characters."
    },
    {
ERR_PAYMENT_MISSING_CDI_KEY,
    "Payment missing Customer district key
\"CDI*\"."
    },
    {
ERR_PAYMENT_MISSING_CID_CLT,
    "Payment Either Customer ID or Last Name
must be entered."
    },
    {
ERR_PAYMENT_MISSING_CID_KEY,
    "Payment missing Customer Key \"CID*\"."
    },
    },

```

```

    {
ERR_PAYMENT_MISSING_CLT_KEY,
    "Payment missing Customer Last Name key
\"CLT*\"."
    },
    {
ERR_PAYMENT_MISSING_CWI_KEY,
    "Payment missing Customer Warehouse key
\"CWI*\"."
    },
    {
ERR_PAYMENT_MISSING_DID_KEY,
    "Payment missing District Key \"DID*\"."
    },
    {
ERR_PAYMENT_MISSING_HAM_KEY,
    "Payment missing Amount key \"HAM*\"."
    },
    {
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
    "Stock Level; missing Threshold key
\"TT*\"."
    },
    {
ERR_STOCKLEVEL_THRESHOLD_INVALID,
    "Stock Level; Threshold value must be in
the range = 1 - 99."
    },
    {
ERR_STOCKLEVEL_THRESHOLD_RANGE,
    "Stock Level Threshold out of
range, range must be 1 - 99."
    },
    {
ERR_VERSION_MISMATCH,
    "Invalid version field. RTE and Web Client
are probably out of sync."
    },
    {
ERR_W_ID_INVALID,
    "Invalid Warehouse ID."
    },
    {
0,
    ""
    },
};
char szTmp[256];
int i = 0;
while (TRUE)
{
    if (errorMsgs[i].szMsg[0] == 0)
    {
        strcpy( szTmp, "Unknown
error number." );
        break;
    }
    if (m_Error ==
errorMsgs[i].iError)
    {

```

```

        strcpy( szTmp,
errorMsgs[i].szMsg );
        break;
    }
    i++;
}
if ( m_szTextDetail)
    strcat( szTmp, m_szTextDetail );
if ( m_SystemErr)
    sprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

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

/* FUNCTION: GetKeyValue
 *
 * PURPOSE:      This function parses a http
formatted string for specific key values.
 *
 * ARGUMENTS:   char
                *pQueryString      http string from client
browser
                char
                *pKey              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

```

```

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

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

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

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

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

    *pQueryString = ptr;
    return atoi(ptr0);

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

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

```

```

*                               is first loaded by the
inet service.
*
*/
void TermInit(void)
{
    EnterCriticalSection(&TermCriticalSection);

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

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

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

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

    LeaveCriticalSection(&TermCriticalSection);
}
/* FUNCTION: TermDeleteAll
*
* PURPOSE: This function frees allocated
resources associated with the terminal structure.
*
* ARGUMENTS: none
*
* RETURNS: None
*
* COMMENTS: This function is called only when
the inet service unloads the TPCC.DLL
*
*/
void TermDeleteAll(void)
{
    EnterCriticalSection(&TermCriticalSection);

    for(int i=1; i<Term.iNumEntries; i++)

```

```

{
    if (Term.pClientData[i].iNextFree
== -1)
        delete
Term.pClientData[i].pTxn;
}

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

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

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

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

        Term.pClientData[iNewTerm].iNextFree = -1;
        // indicates this position is in use
    }
    else
    {
        // no open slots, so find the
slot that hasn't been used in the longest time and
reuse it
        for(iNewTerm=1, i=1,
iTickCount=0x7FFFFFFF; i<Reg.dwMaxConnections; i++)
        {
            if (iTickCount >
Term.pClientData[i].iTickCount)
            {
                iTickCount =
Term.pClientData[i].iTickCount;
                iNewTerm = i;
            }
        }
        // if oldest term is less than
one minute old, it probably means that more
connections

```

```

// are being attempted than were
specified as "Max Connections" at install. In this
case,
// do not bump existing
connection; instead, return error to requester.
if ((GetTickCount() - iTickCount)
< 60000)
{
    LeaveCriticalSection(&TermCriticalSection);
    throw new CWEBCLNT_ERR(
ERR_MAX_CONNECTIONS_EXCEEDED );
}

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

LeaveCriticalSection(&TermCriticalSection);
return iNewTerm;
}
/* FUNCTION: TermDelete
*
* PURPOSE: This function makes a terminal
entry in the Term array available for reuse.
*
* ARGUMENTS: int id
Terminal id of client exiting
*
*/
void TermDelete(int id)
{
    if ( id > 0 && id < Term.iNumEntries )
    {
        delete Term.pClientData[id].pTxn;
        // put onto free list

        EnterCriticalSection(&TermCriticalSection);

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

        LeaveCriticalSection(&TermCriticalSection);
    }
}
/* FUNCTION: MakeErrorForm
*/
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int
iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer )
{

```

```

wsprintf(szBuffer,
"HTML<HEAD><TITLE>TPC-C
Error</TITLE></HEAD><BODY>"
"FORM ACTION=\"tpcc.dll\"
METHOD=\"GET\">"
"INPUT TYPE=\"hidden\"
NAME=\"STATUSID\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\"
NAME=\"ERROR\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\"
NAME=\"FORMID\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\"
NAME=\"TERMINID\" VALUE=\"%d\">"
"INPUT TYPE=\"hidden\"
NAME=\"SYCID\" VALUE=\"%d\">"
"An Error
Occurred</BOLD><BR><BR>"
"ss"
"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=\"SYCID\" 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=\"SYCID\" 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>"
"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></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>";
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\">"
                "<INPUT TYPE=\"submit\">"
                "</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:
                W_tax: %5.2f D_tax: %5.2f <BR> <BR>"
                " Supp_W Item_Id Item Name
                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 );
            }

```



```

        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:  %5.50s<BR>          %-
50.50s<BR>          %5.50s<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>";
    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=\"SYCID\" 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:
Carrier-
<BR>"
            "Supply-W Item-Id
Qty Amount Delivery-Date<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> </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> </font></PRE>\"
            "<CHR><INPUT
TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">\"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">\"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">\"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">\"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">\"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">\"
            "</BODY></FORM></HTML>\"
            , pDeliveryData-
>o_carrier_id,
            (pDeliveryData-
>exec_status_code == eOK) ? "Delivery has been
queued." : "Delivery Post Failed "
            );
    }
}
/* FUNCTION: ProcessNewOrderForm
 *
 * PURPOSE:      This function gets and validates
the input data from the new order form
 *                filling in the required
input variables. it then calls the SQLNewOrder

```

```

 *                transaction, constructs
the output form and writes it back to client
 *                browser.
 */
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer)
{
    PNEW_ORDER_DATA    pNewOrder;
    pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();
    ZeroMemory(pNewOrder,
sizeof(NEW_ORDER_DATA));
    pNewOrder->w_id =
Term.pClientData[iTermId].w_id;
    GetNewOrderData(pECB->lpszQueryString,
pNewOrder);
    Term.pClientData[iTermId].pTxn->NewOrder();
    pNewOrder = Term.pClientData[iTermId].pTxn-
>BuffAddr_NewOrder();
    MakeNewOrderForm(iTermId, pNewOrder,
OUTPUT_FORM, szBuffer );
}
/* FUNCTION: void ProcessPaymentForm
 *
 * PURPOSE:      This function gets and validates
the input data from the payment form
 *                filling in the required
input variables. It then calls the SQLPayment
 *                transaction, constructs
the output form and writes it back to client
 *                browser.
 *
 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
                *pECB    passed in structure pointer from
inetsrv.
                int
                iTermId    client browser terminal id
 */
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer)
{
    PPAYMENT_DATA    pPayment;
    pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();
    ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
    pPayment->w_id =
Term.pClientData[iTermId].w_id;
    GetPaymentData(pECB->lpszQueryString,
pPayment);
    Term.pClientData[iTermId].pTxn->Payment();

```

```

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

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

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

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

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

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

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

```

```

 *              int
 *              iTermId  client browser terminal id
 */

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

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

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

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

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

/* FUNCTION: ProcessStockLevelForm
 *
 * PURPOSE:      This function gets and validates
the input data from the Stock Level
 *              form filling in the
required input variables. It then calls the
 *              SQLStockLevel
transaction, constructs the output form and writes it
 *              back to client browser.
 *

```

```

 * ARGUMENTS:    EXTENSION_CONTROL_BLOCK
 *pECB          passed in structure pointer from
inetsrv.
 *              int
 *              iTermId  client browser terminal id
 */

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

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

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

    pStockLevel->threshold =
GetIntKeyValue(&ptr, "TT*");
    ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
    ERR_STOCKLEVEL_THRESHOLD_INVALID);
    if ( pStockLevel->threshold >= 100 ||
pStockLevel->threshold < 0 )
        throw new CWBCLNT_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]
=
    "SP03*", "SP04*",      { "SP00*", "SP01*", "SP02*",
"SP08*", "SP09*",        "SP05*", "SP06*", "SP07*",
"SP13*", "SP14*" };
    static char
szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "IID00*", "IID01*", "IID02*",
"IID03*", "IID04*",
"IID08*", "IID09*",    "IID05*", "IID06*", "IID07*",
"IID13*", "IID14*" };
    static char
szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
    { "Qty00*", "Qty01*", "Qty02*",
"Qty03*", "Qty04*",    "Qty05*", "Qty06*", "Qty07*",
"Qty08*", "Qty09*",    "Qty10*", "Qty11*", "Qty12*",
"Qty13*", "Qty14*" };

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

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

        ol_i_id =
pNewOrderData->OL[items].ol_i_id =
        GetIntKeyValue(&ptr, szIID[i],
ERR_NEWORDER_MISSING_IID_KEY,
ERR_NEWORDER_ITEMID_INVALID);
        if ( ol_i_id > 999999
|| ol_i_id < 1 )

```

```

throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE );

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

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

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

    pNewOrderData->o_ol_cnt = items;
}

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

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

```

```

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

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

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

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

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

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

```

```

                                throw new CWBCLNT_ERR(
ERR_PAYMENT_CID_AND_CLT );
    }

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

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

    pOrderStatusData->d_id =
GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_ORDERSTATUS_DID_INVALID);

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

        strcpy( szTmp );
        if ( strlen(szTmp) >
LAST_NAME_LEN )
            throw new CWBCLNT_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 CWBCLNT_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 CWBCLNT_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 *dotp;
    BOOL bValid;

    if ( *ptr == 0 )
        return FALSE;

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

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

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

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

```

tpcc.def

LIBRARY TPCC.DLL

EXPORTS

```

    GetExtensionVersion @1
    HttpExtensionProc @2
    TerminateExtension @3

```

tpcc.h

```

/* FILE:      TPCC.H
 *
 * Microsoft
TPC-C Kit Ver. 4.20.000
 *
 * Copyright
Microsoft, 1999
 *
 * All Rights Reserved
 *
 * Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99

```

```

*
* PURPOSE: Header file for ISAPI TPCC.DLL,
defines structures and functions used in the isapi
tpcc.dll.
*/
//VERSION RESOURCE DEFINES
#define _APS_NEXT_RESOURCE_VALUE
        101
#define _APS_NEXT_COMMAND_VALUE
        40001
#define _APS_NEXT_CONTROL_VALUE
        1000
#define _APS_NEXT_SYMED_VALUE
        101

#define TP_MAX_RETRIES
        50

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

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

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

```

```

        int
                d_id;
        //district id
assigned at welcome form

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

        CTPCC_BASE
                *pTxn;

} CLIENTDATA, *PCLIENTDATA;

//This structure is used to define the operational
interface for terminal id support
typedef struct _TERM
{
        int
                iNumEntries;

        //total allocated terminal array entries
        int
                iFreeList;

        //next available terminal array element or
-1 if none
        int
                iMasterSyncId;
        //synchronization id
        CLIENTDATA
                *pClientData;
        //pointer to
allocated client data
} TERM;

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

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

```

```

        ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
        ERR_NEWORDER_MISSING_IID_KEY,
        ERR_NEWORDER_MISSING_QTY_KEY,
        ERR_NEWORDER_MISSING_SUPPW_KEY,
        ERR_NEWORDER_NOITEMS_ENTERED,
        ERR_NEWORDER_QTY_INVALID,
        ERR_NEWORDER_QTY_RANGE,
        ERR_NEWORDER_QTY_WITHOUT_SUPPW,
        ERR_NEWORDER_SUPPW_INVALID,
        ERR_NO_SERVER_SPECIFIED,
        ERR_ORDERSTATUS_CID_AND_CLT,
        ERR_ORDERSTATUS_CID_INVALID,
        ERR_ORDERSTATUS_CLT_RANGE,
        ERR_ORDERSTATUS_DID_INVALID,
        ERR_ORDERSTATUS_MISSING_CID_CLT,
        ERR_ORDERSTATUS_MISSING_CID_KEY,
        ERR_ORDERSTATUS_MISSING_CLT_KEY,
        ERR_ORDERSTATUS_MISSING_DID_KEY,
        ERR_PAYMENT_CDI_INVALID,
        ERR_PAYMENT_CID_AND_CLT,
        ERR_PAYMENT_CUSTOMER_INVALID,
        ERR_PAYMENT_CWI_INVALID,
        ERR_PAYMENT_DISTRICT_INVALID,
        ERR_PAYMENT_HAM_INVALID,
        ERR_PAYMENT_HAM_RANGE,
        ERR_PAYMENT_LAST_NAME_TO_LONG,
        ERR_PAYMENT_MISSING_CDI_KEY,
        ERR_PAYMENT_MISSING_CID_CLT,
        ERR_PAYMENT_MISSING_CID_KEY,
        ERR_PAYMENT_MISSING_CLT,
        ERR_PAYMENT_MISSING_CLT_KEY,
        ERR_PAYMENT_MISSING_CWI_KEY,
        ERR_PAYMENT_MISSING_DID_KEY,
        ERR_PAYMENT_MISSING_HAM_KEY,

        ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        ERR_STOCKLEVEL_THRESHOLD_RANGE,
        ERR_VERSION_MISMATCH,
        ERR_W_ID_INVALID
};

class CWEBCLNT_ERR : public CBaseErr
{
public:
        CWEBCLNT_ERR(WEBERROR Err)
        {
                m_Error = Err;
                m_szTextDetail = NULL;
                m_SystemErr = 0;
                m_szErrorText = NULL;
        };

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

```

```

        m_SystemErr =
dwSystemErr;
        m_szErrorText = NULL;
    };
    ~CWEBCLNT_ERR()
    {
        if (m_szTextDetail !=
NULL)
            delete []
m_szTextDetail;
        if (m_szErrorText !=
NULL)
            delete []
m_szErrorText;
    };
    WEBERROR m_Error;
    char
*m_szTextDetail; //
    char
*m_szErrorText;
    DWORD
        m_SystemErr;

    int ErrorType() {return
ERR_TYPE_WEBDLL;};
    char *ErrorTypeStr() { return
"WEBCLIENT"; }

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

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

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD
ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd, int *pFormId, int *pTermId, int
*pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int
iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char
*szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int
iError, int iErrorType, char *szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey,
char *pValue, int iMax, WEBERROR err);

```

```

int GetIntKeyValue(char **pQueryString, char *pKey,
WEBERROR NoKeyErr, WEBERROR NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int
iType, int iErrorNum, int iTermId, int iSyncId, char
*szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char
*szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA
*pStockLevelData, BOOL bInput, char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA
*pNewOrderData, BOOL bInput, char *szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA
*pPaymentData, BOOL bInput, char *szForm);
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput,
char *szForm);
void MakeDeliveryForm(int iTermId, DELIVERY_DATA
*pDeliveryData, BOOL bInput, char *szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK
*pECB, int iTermId, char *szBuffer);
void GetNewOrderData(LPSTR lpszQueryString,
NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString,
PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString,
ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
// Separate function to be able to use Win32
exception handling in
// HttpExtensionProc.
void ProcessCommand(EXTENSION_CONTROL_BLOCK *pECB,
char* szBuffer, int& TermId, int& iSyncId);

```

tpcc.rc

```

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

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

```

```

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

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

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

#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"
    BEGIN
        BLOCK "040904b0"
        BEGIN
            VALUE "Comments", "TPC-C HTML DLL
Server\0"
            VALUE "CompanyName", "Microsoft\0"
            VALUE "FileDescription", "TPC-C HTML DLL
Server\0"
            VALUE "FileVersion", "0, 4, 0, 0\0"
            VALUE "InternalName", "tpcc\0"
            VALUE "LegalCopyright", "Copyright ©
1997\0"
            VALUE "OriginalFilename", "tpcc.dll\0"
            VALUE "ProductName", "Microsoft tpcc\0"
            VALUE "ProductVersion", "0, 4, 0, 0\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
#endif // !_MAC

```

```

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

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

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

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

#endif // APSTUDIO_INVOKED

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

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

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

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

```

```

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//

////////////////////////////////////
////////////////////////////////////
#endif // not APSTUDIO_INVOKED


```

tpcc_com.cpp

```

/* FILE: TPC_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 "..\..\common\src\tpcc_com_errorcode.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
DllDecl CTPCC_COM* CTPCC_COM_new(BOOL
bSinglePool)

```

```

{
    return new CTPCC_COM(bSinglePool);
}

CTPCC_COM::CTPCC_COM(BOOL bSinglePool)
{
    HRESULT hr = NULL;
    long lRet = 0;
    ULONG ulTmpSize = 0;

    m_pTxn = NULL;
    m_pNewOrder = NULL;
    m_pPayment = NULL;
    m_pStockLevel = NULL;
    m_pOrderStatus = NULL;

    m_bSinglePool = bSinglePool;

    ulTmpSize = (ULONG) sizeof(COM_DATA);
    VariantInit(&m_vTxn);
    m_vTxn.vt = VT_SAFEARRAY;

    m_vTxn.parray =
    SafeArrayCreateVector(VT_UI1, ulTmpSize, ulTmpSize);
    if (!m_vTxn.parray)
        throw new CCOMERR( E_FAIL );

    memset((void*)m_vTxn.parray-
    >pvData,0,ulTmpSize);
    m_pTxn = (COM_DATA*)m_vTxn.parray->pvData;

    hr = CoInitializeEx(NULL,
    COINIT_MULTITHREADED);
    if (FAILED(hr))
    {
        throw new CCOMERR( hr );
    }

    // create components
    if (m_bSinglePool)
    {
        hr = CoCreateInstance(CLSID_TPCC,
        NULL, CLSCTX_SERVER, IID_ITPCC, (void
        **)&m_pNewOrder);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        // all txns will use same
        component
        {
            m_pPayment = m_pNewOrder;
            m_pStockLevel = m_pNewOrder;
            m_pOrderStatus = m_pNewOrder;
        }
        else
        {
            // use different components for
            each txn
            {
                hr =
                CoCreateInstance(CLSID_NewOrder, NULL, CLSCTX_SERVER,
                IID_ITPCC, (void **)&m_pNewOrder);
                if (FAILED(hr))
                    throw new CCOMERR(hr);
            }
        }
    }
}

```



```

        hr =
CoCreateInstance(CLSID_Payment, NULL, CLSCTX_SERVER,
IID_ITPCC, (void **)&m_pPayment);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr =
CoCreateInstance(CLSID_StockLevel, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pStockLevel);
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr =
CoCreateInstance(CLSID_OrderStatus, NULL,
CLSCTX_SERVER, IID_ITPCC, (void **)&m_pOrderStatus);
        if (FAILED(hr))
            throw new CCOMERR(hr);
    }

    // call setcomplete to release each
    component back into pool
    hr = m_pNewOrder->CallSetComplete();
    if (FAILED(hr))
        throw new CCOMERR(hr);

    if (!m_bSinglePool)
    {
        hr = m_pPayment-
>CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = m_pStockLevel-
>CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);

        hr = m_pOrderStatus-
>CallSetComplete();
        if (FAILED(hr))
            throw new CCOMERR(hr);
    }
}

CTPCC_COM::~CTPCC_COM()
{
    if (m_pTxn)
        SafeArrayDestroy(m_vTxn.parray);

    ReleaseInterface(m_pNewOrder);
    if (!m_bSinglePool)
    {
        ReleaseInterface(m_pPayment);
        ReleaseInterface(m_pStockLevel);
        ReleaseInterface(m_pOrderStatus);
    }
    CoUninitialize();
}

void CTPCC_COM::NewOrder()
{
    VARIANT                vTxn_out;

```

```

        HRESULT hr = m_pNewOrder->NewOrder(m_vTxn,
&vTxn_out);

        if (FAILED(hr) && hr != E_TPCCCOM)
            throw new CCOMERR( hr ); //
COM call didn't succeed and there is no output
structure

        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        hr = SafeArrayDestroy(vTxn_out.parray);
        if (hr != S_OK)
            throw new CCOMERR( hr );

        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) && hr != E_TPCCCOM)
        throw new CCOMERR( hr ); //
COM call didn't succeed and there is no output
structure

        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        hr = SafeArrayDestroy(vTxn_out.parray);
        if (hr != S_OK)
            throw new CCOMERR( hr );

        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) && hr != E_TPCCCOM)
        throw new CCOMERR( hr ); //
COM call didn't succeed and there is no output
structure

        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        hr = SafeArrayDestroy(vTxn_out.parray);
        if (hr != S_OK)
            throw new CCOMERR( hr );

        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) && hr != E_TPCCCOM)
        throw new CCOMERR( hr ); //
COM call didn't succeed and there is no output
structure

        memcpy(m_pTxn, (void *)vTxn_out.parray-
>pvData,vTxn_out.parray->rgsabound[0].cElements);
        hr = SafeArrayDestroy(vTxn_out.parray);
        if (hr != S_OK)
            throw new CCOMERR( hr );

        if ( m_pTxn->ErrorType != ERR_SUCCESS )
            throw new CCOMERR( m_pTxn-
>ErrorType, m_pTxn->error );
    }
}

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()
    {
        if (m_iErrorType == 0)
            return m_hr;
        // return COM error
        else
            return
m_iError; // return impersonated error
    }

    char *ErrorText()
    {
        if (m_hr == S_OK)
            sprintf(
m_szErrorText, "Error: Class %d, error # %d",
m_iErrorType, m_iError );
        else
            sprintf(
m_szErrorText, "Error: COM HRESULT %x", m_hr );
    }

```

```

        return m_szErrorText;
    }
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

// COM Interface pointers
    ITPCC*
m_pNewOrder;
    ITPCC*
m_pPayment;
    ITPCC*
m_pStockLevel;
    ITPCC*
m_pOrderStatus;

    struct COM_DATA
    {
        int ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA NewOrder;
            PAYMENT_DATA Payment;
            DELIVERY_DATA Delivery;
            STOCK_LEVEL_DATA StockLevel;
            ORDER_STATUS_DATA OrderStatus;
        } u;
    } *m_pTxn;

    VARIANT m_vTxn;

public:
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

    inline PNEW_ORDER_DATA
BuffAddr_NewOrder() { return
&m_pTxn->u.NewOrder; };
    inline PPAYMENT_DATA
BuffAddr_Payment() { return
&m_pTxn->u.Payment; };
    inline PDELIVERY_DATA
BuffAddr_Delivery() { return
&m_pTxn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel() { return
&m_pTxn->u.StockLevel; };
    inline PORDER_STATUS_DATA
BuffAddr_OrderStatus() { return
&m_pTxn->u.OrderStatus; };

    void NewOrder ();
    void Payment ();
    void StockLevel ();
    void OrderStatus ();
}

```

```

void Delivery ();
{ throw new CCOMERR(E_NOTIMPL); } // not supported
};

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM*
CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM) (BOOL);

tpcc_com_all.
cpp
/* FILE: TPC_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 class.
* Contact: Charles Levine
(clevine@microsoft.com)
*
* Change history:
* 4.20.000 - updated rev number to
match kit
*/

#define STRICT
#define WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

#include <stdio.h>
#include <atlbase.h>
//You may derive a class from CComModule and use it
if you want to override
//something, but do not change the name of _Module
extern CComModule _Module;

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>

```

```

//#include <atlimpl.cpp>
#include <comsvcs.h>

#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"
//tpckit transaction
header contains definations of structures specific to
TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\common\src\tpcc_com_errorcode.h"
#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_ODBC        *pCTPCC_ODBC_new;

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

////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD
dwReason, LPVOID /*lpReserved*/)
{
    char szDllName[128];

```

```

        try
        {
            if (dwReason ==
DLL_PROCESS_ATTACH)
            {
                _Module.Init(ObjectMap,
hInstance);

                DisableThreadLibraryCalls(hInstance);

                DWORD dwSize =
MAX_COMPUTERNAME_LENGTH+1;

                GetComputerName(szMyComputerName, &dwSize);

                szMyComputerName[dwSize] = 0;

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

                if (Reg.eDB_Protocol ==
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(&hConnectCritical
Section);

                    }

                }
            }
            else if (dwReason ==
DLL_PROCESS_DETACH)

```

```

                _Module.Term();

            }
            catch (CBaseErr *e)
            {
                TCHAR szMsg[256];

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

                delete e;
                return FALSE;
            }
            catch (...)
            {
                WriteMessageToEventLog(TEXT("Unhandled
exception in object DllMain"));
                return FALSE;
            }

            return TRUE; // OK

}

////////////////////////////////////
// Used to determine whether the DLL can be unloaded
by OLE

STDAPI DllCanUnloadNow(void)
{
    return (_Module.GetLockCount()==0) ? S_OK :
S_FALSE;
}

////////////////////////////////////
// Returns a class factory to create an object of the
requested type

STDAPI DllGetClassObject(REFCLSID rclsid, REFIID
riid, LPVOID* ppv)
{
    return _Module.GetClassObject(rclsid, riid,
ppv);
}

////////////////////////////////////
// DllRegisterServer - Adds entries to the system
registry

STDAPI DllRegisterServer(void)
{
    // registers object, typelib and all
interfaces in typelib
    return _Module.RegisterServer(TRUE);
}

```

```

////////////////////////////////////
////////////////////////////////////
// DllUnregisterServer - Removes entries from the
system registry

STDAPI DllUnregisterServer(void)
{
    _Module.UnregisterServer();
    return S_OK;
}

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

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

    _sprintf(szMsg, TEXT("Error in COM+ TPC-C
Component: "));
    lpszStrings[0] = szMsg;
    lpszStrings[1] = lpszMsg;

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

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
*/
char* CCOMPONENT_ERR::ErrorText(void)

```

```

{
    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,
"Required entries missing from registry."
        },
        { ERR_LOADDLL_FAILED,
"Load of DLL failed. DLL="
        },
        { ERR_GETPROCADDR_FAILED,
"Could not map proc in DLL. GetProcAddress
error. DLL="
        },
        { ERR_UNKNOWN_DB_PROTOCOL,
"Unknown database protocol specified in
registry."
        },
        { 0, ""
        }
    };

    char szTmp[256];
    int i = 0;
    while (TRUE)
    {
        if (errorMsgs[i].szMsg[0] == 0)
        {
            strcpy( szTmp, "Unknown
error number." );
            break;
        }
        if (m_Error ==
errorMsgs[i].iError)
        {
            strcpy( szTmp,
errorMsgs[i].szMsg );
            break;
        }
        i++;
    }
    if (m_szTextDetail)
        strcat( szTmp, m_szTextDetail );
    if (m_SystemErr)
        sprintf( szTmp+strlen(szTmp), "
Error=%d", m_SystemErr );

    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(&hConnectCriticalSecti
on);

    Sleep(Reg.dwConnectDelay);

    LeaveCriticalSection(&hConnectCriticalSecti
on);
}

if (m_pTxn)
{
    delete m_pTxn;
}

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext(
IID_IObjectContext, (void **)&pObjectContext );
    pObjectContext->SetComplete();
    ReleaseInterface(pObjectContext);
    return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch *
pUnk)
{
    // Code to access construction string, if
needed later...
    //
    if (!pUnk)
        return E_UNEXPECTED;
    //
    IObjectConstructString * pString
= NULL;
    //
    HRESULT hr = pUnk-
>QueryInterface(IID_IObjectConstructString, (void
**)&pString);
    //
    pString->Release();

    try
    {
        // Pace connection creation for
VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {
            EnterCriticalSection(&hConnectCriticalSecti
on);

            Sleep(Reg.dwConnectDelay);

```

```

    LeaveCriticalSection(&hConnectCriticalSection);
}

if (Reg.eDB_Protocol == ODBC)
    m_pTxn =
pCTPCC_ODBC_new(
    Reg.szDbServer, Reg.szDbUser,
    Reg.szDbPassword,

    szMyComputerName, Reg.szDbName,

    Reg.szSPPrefix,
    Reg.bCallNoDuplicatesNewOrder);
}
catch (CBaseErr *e)
{
    TCHAR szMsg[256];
    _sntprintf(szMsg, sizeof(szMsg),
"%s error in CTPCC_Common::Construct, code %d: %s",
    e-
>ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
    WriteMessageToEventLog( szMsg );
    delete e;
    return E_FAIL;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled
exception in object ::Construct"));
    return E_FAIL;
}

return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in,
VARIANT* txn_out)
{
    PNEW_ORDER_DATA    pNewOrder;
    COM_DATA            *pData;
    COM_DATA            *pOutData;

    try
    {
        // Allocate output structure
        first because it is also used in the catch clauses.
        //
        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

        txn_in.parray->rgsabound-
>cElements,

        txn_in.parray->rgsabound-
>cElements);
    }
}

```

```

        if (txn_out->parray == NULL) //
sanity error checking - for very rare case, but to be
sure
        {
            return E_OUTOFMEMORY;
        }

        pOutData = (COM_DATA*)txn_out-
>parray->pvData;

        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

        memcpy( &pOutData->u.NewOrder,
pNewOrder, sizeof(NEW_ORDER_DATA));

        pOutData->retval = ERR_SUCCESS;
        pOutData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes,
        component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
            m_bCanBePooled = FALSE;

        pOutData->retval = e-
>ErrorType();

        pOutData->error = e->ErrorNum();
        delete e;
        return E_TPCCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception in CTPCC_Common::NewOrder.));
        pOutData->retval =
ERR_TYPE_LOGIC;

        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCCOM;
    }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in,
VARIANT* txn_out)
{
    PPAYMENT_DATA    pPayment;
    COM_DATA          *pData;
    COM_DATA          *pOutData;

    try

```

```

    {
        // Allocate output structure
        first because it is also used in the catch clauses.
        //
        VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

        txn_in.parray->rgsabound-
>cElements,

        txn_in.parray->rgsabound-
>cElements);

        if (txn_out->parray == NULL) //
sanity error checking - for very rare case, but to be
sure
        {
            return E_OUTOFMEMORY;
        }

        pOutData = (COM_DATA*)txn_out-
>parray->pvData;

        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

        memcpy( &pOutData->u.Payment,
pPayment, sizeof(PAYMENT_DATA));

        pOutData->retval = ERR_SUCCESS;
        pOutData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes,
        component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
            m_bCanBePooled = FALSE;

        pOutData->retval = e-
>ErrorType();

        pOutData->error = e->ErrorNum();
        delete e;
        return E_TPCCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception in CTPCC_Common::Payment.));
        pOutData->retval =
ERR_TYPE_LOGIC;

```

```

        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCCOM;
    }
}

HRESULT CTPCC_Common::StockLevel(VARIANT txn_in,
VARIANT* txn_out)
{
    PSTOCK_LEVEL_DATA  pStockLevel;
    COM_DATA            *pData;
    COM_DATA            *pOutData;

    try
    {
        // Allocate output structure
        // first because it is also used in the catch clauses.
        // VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

        txn_in.parray->rgsabound-
>cElements,

        txn_in.parray->rgsabound-
>cElements);
        if (txn_out->parray == NULL) //
sanity error checking - for very rare case, but to be
sure
        {
            return E_OUTOFMEMORY;
        }

        pOutData = (COM_DATA*)txn_out-
>parray->pvData;

        pData = (COM_DATA*)txn_in.parray-
>pvData;
        pStockLevel = m_pTxn-
>BuffAddr_StockLevel();

        memcpy(pStockLevel, &pData-
>u.StockLevel, sizeof(STOCK_LEVEL_DATA));

        m_pTxn->StockLevel();

        memcpy( &pOutData->u.StockLevel,
pStockLevel, sizeof(STOCK_LEVEL_DATA));

        pOutData->retval = ERR_SUCCESS;
        pOutData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes, component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
            m_bCanBePooled = FALSE;
    }
}

```

```

        pOutData->retval = e-
>ErrorType();
        pOutData->error = e->ErrorNum();
        delete e;
        return E_TPCCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception in CTPCC_Common::StockLevel."););
        pOutData->retval =
ERR_TYPE_LOGIC;
        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCCOM;
    }
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in,
VARIANT* txn_out)
{
    PORDER_STATUS_DATA pOrderStatus;
    COM_DATA            *pData;
    COM_DATA            *pOutData;
    try
    {
        // Allocate output structure
        // first because it is also used in the catch clauses.
        // VariantInit(txn_out);
        txn_out->vt = VT_SAFEARRAY;
        txn_out->parray =
SafeArrayCreateVector( VT_UI1,

        txn_in.parray->rgsabound-
>cElements,

        txn_in.parray->rgsabound-
>cElements);
        if (txn_out->parray == NULL) //
sanity error checking - for very rare case, but to be
sure
        {
            return E_OUTOFMEMORY;
        }

        pOutData = (COM_DATA*)txn_out-
>parray->pvData;

        pData = (COM_DATA*)txn_in.parray-
>pvData;
        pOrderStatus = m_pTxn-
>BuffAddr_OrderStatus();

        memcpy(pOrderStatus, &pData-
>u.OrderStatus, sizeof(ORDER_STATUS_DATA));

        m_pTxn->OrderStatus();

        memcpy( &pOutData->u.OrderStatus,
pOrderStatus, sizeof(ORDER_STATUS_DATA));
    }
}

```

```

        pOutData->retval = ERR_SUCCESS;
        pOutData->error = 0;
        return S_OK;
    }
    catch (CBaseErr *e)
    {
        // check for lost database
        connection; if yes, component is toast
        if ( ((e->ErrorType() ==
ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
            m_bCanBePooled = FALSE;
        pOutData->retval = e-
>ErrorType();
        pOutData->error = e->ErrorNum();
        delete e;
        return E_TPCCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
exception in CTPCC_Common::OrderStatus."););
        pOutData->retval =
ERR_TYPE_LOGIC;
        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCCOM;
    }
}

```

tpcc_com_all.def

; tpcc_com_all.def : Declares the module parameters.

```

LIBRARY      "tpcc_com_all.dll"

EXPORTS
    DllCanUnloadNow      PRIVATE
    DllGetClassObject    PRIVATE
    DllRegisterServer    PRIVATE
    DllUnregisterServer  PRIVATE

```

tpcc_com_all.h

/* this ALWAYS GENERATED file contains the
definitions for the interfaces */

```

/* File created by MIDL compiler version 6.00.0361
*/
/* at Thu Mar 16 18:21:15 2006
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
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( )

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

/* verify that the <rpcndr.h> version is high enough
to compile this file*/
#ifndef REQUIRED_RPCNDR_H_VERSION
#define REQUIRED_RPCNDR_H_VERSION__ 475
#endif

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

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

#ifndef tpcc_com_all_h_
#define tpcc_com_all_h_

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef TPCC_FWD_DEFINED__
#define TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif /* TPCC_FWD_DEFINED__ */

#ifndef NewOrder_FWD_DEFINED__
#define NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif /* NewOrder_FWD_DEFINED__ */

#ifndef OrderStatus_FWD_DEFINED__
#define OrderStatus_FWD_DEFINED__

```

```

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif /* OrderStatus_FWD_DEFINED__ */

#ifndef Payment_FWD_DEFINED__
#define Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif /* Payment_FWD_DEFINED__ */

#ifndef StockLevel_FWD_DEFINED__
#define StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else
typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif /* StockLevel_FWD_DEFINED__ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

extern RPC_IF_HANDLE
_MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
_MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#endif /* 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 /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_all_i. C

```
/* 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 6.00.0361
*/
/* at Thu Mar 16 18:21:15 2006
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
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)

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
```

```
DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00
,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0x11D2,0xBA,0x4E,0x00
,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
```

```
}
#endif

#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

/* 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 6.00.0361
*/
/* at Thu Mar 16 18:21:15 2006
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win64 (32b run,appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum
stub_data
VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AMD64)

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

#ifdef __cplusplus
extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \

DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)
```



```

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char c[8];
} IID;

#endif // __IID_DEFINED__

#ifndef CLSID_DEFINED
#define CLSID_DEFINED
typedef IID CLSID;
#endif // CLSID_DEFINED

#define
MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
b7,b8) \
    const type name =
{1,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,
0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0x
C0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,0x0
0,0xC0,0x4F,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,0x11D2,0xBA,0x4E,0x00
,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0
x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif

```

```

#endif /* defined(_M_IA64) || defined(_M_AMD64) */

```

tpcc_com_errorcode.h

```

/* FILE: TPCC_COM_ERRORCODE.H
 * Microsoft
 * TPC-C Kit Ver. 4.20.000
 * Copyright
 * Microsoft, 1999
 * All Rights Reserved
 *
 * not yet
 * audited
 *
 * PURPOSE: Header file defining the error
code returned from ITPCC COM interface.
 *
 * Change history:
 * 4.20.000 - first version
 */

// Error return value for methods in ITPCC interface.
//
// Define as 0x80042345 (decimal -2147212475 ).
//
const HRESULT E_TPCCCOM = MAKE_HRESULT
(SEVERITY_ERROR, FACILITY_ITF, 0x2345);

```

tpcc_com_ps.def

```

LIBRARY "tpcc_com_ps"

EXPORTS
    DllGetClassObject PRIVATE
    DllCanUnloadNow PRIVATE
    GetProxyDllInfo PRIVATE
    DllRegisterServer PRIVATE
    DllUnregisterServer PRIVATE

```

tpcc_com_ps.h

```

/* this ALWAYS GENERATED file contains the
definitions for the interfaces */

```

```

/* File created by MIDL compiler version 6.00.0361
 */
/* at Thu Mar 16 18:21:12 2006

```

```

*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, Wl, 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( )

#pragma warning( disable: 4049 ) /* more than 64k
source lines */

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

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

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

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

#ifndef _tpcc_com_ps_h_
#define _tpcc_com_ps_h_

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef _ITPCC_FWD_DEFINED_
#define _ITPCC_FWD_DEFINED_
typedef interface ITPCC ITPCC;
#endif /* _ITPCC_FWD_DEFINED_ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface _MIDL_itf_tpcc_com_ps_0000 */

```

```

/* [local] */

extern RPC_IF_HANDLE
__MIDL_itf_tpsc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpsc_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("FEEB6AA2-84B1-11d2-BA47-00C04FBFE08B")
    ITPCC : public IUnknown
    {
    public:
        virtual HRESULT STDMETHODCALLTYPE NewOrder(
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT *txn_out) = 0;

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

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

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

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

        virtual HRESULT STDMETHODCALLTYPE CallSetComplete(
            void) = 0;
    };

#else /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

        HRESULT ( STDMETHODCALLTYPE *QueryInterface
        )(
            ITPCC * This,
            /* [in] */ REFIID riid,
            /* [iid_is][out] */ void **ppvObject);

```

```

        ULONG ( STDMETHODCALLTYPE *AddRef ) (
            ITPCC * This);

        ULONG ( STDMETHODCALLTYPE *Release ) (
            ITPCC * This);

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

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

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

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

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

        HRESULT ( STDMETHODCALLTYPE *CallSetComplete ) (
            ITPCC * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl *lpVtbl;
    };

#ifdef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
    (This)->lpVtbl->QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
    (This)->lpVtbl->AddRef(This)

#define ITPCC_Release(This) \
    (This)->lpVtbl->Release(This)

#define ITPCC_NewOrder(This,txn_in,txn_out) \
    (This)->lpVtbl->NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out) \

```

```

        (This)->lpVtbl->Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out) \
    (This)->lpVtbl->Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out) \
    (This)->lpVtbl->StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
    (This)->lpVtbl->OrderStatus(This,txn_in,txn_out)

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

#endif /* COBJMACROS */

#endif /* C style interface */

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

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

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

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *pdwStubPhase);

HRESULT STDMETHODCALLTYPE 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 STDMETHODCALLTYPE ITPCC_StockLevel_Proxy(
    ITPCC * This,

```

```

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

void __RPC_STUB ITPCC_StockLevel_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

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

void __RPC_STUB ITPCC_OrderStatus_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
IRpcStubBuffer *This,
IRpcChannelBuffer *pRpcChannelBuffer,
PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long             __RPC_USER
VARIANT_UserSize(        unsigned long *, unsigned long
, VARIANT * );
unsigned char * __RPC_USER VARIANT_UserMarshal(
unsigned long *, unsigned char *, VARIANT * );
unsigned char * __RPC_USER
VARIANT_UserUnmarshal(unsigned long *, unsigned char
*, VARIANT * );
void                    __RPC_USER
VARIANT_UserFree(        unsigned long *, VARIANT * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif
#endif

```

tpcc_com_ps. idl

```

/* FILE: ITPCC.IDL
* Microsoft
TPC-C Kit Ver. 4.20.000
* Copyright
Microsoft, 1999
* All Rights Reserved
*
* not yet
audited
*
* PURPOSE: Defines the interface used by
TPCC. This interface can be implemented by C++
components.
*
* Change history:
* 4.20.000 - first version
*/

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

[
    object,
    oleautomation,
    uuid(FEEE6AA2-84B1-11d2-BA47-
00C04FBFE08B),
    helpstring("ITPCC Interface"),
    pointer_default(unique)
]
interface ITPCC : IUnknown
{
    HRESULT __stdcall NewOrder(
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        )
    );
    HRESULT __stdcall Payment(
        (
            [in] VARIANT txn_in,
            [out] VARIANT *txn_out
        )
    );
    HRESULT __stdcall Delivery(
        (

```

```

[in] VARIANT txn_in,
[out] VARIANT *txn_out
);
);
HRESULT __stdcall StockLevel(
(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);
);
HRESULT __stdcall OrderStatus(
(
[in] VARIANT txn_in,
[out] VARIANT *txn_out
);
);
HRESULT __stdcall CallSetComplete(
(
);
); // interface ITPCC

```

tpcc_com_ps_i .c

```

/* 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 6.00.0361
*/
/* at Thu Mar 16 18:21:12 2006
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, Wl, 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)

    #pragma warning( disable: 4049 ) /* more than 64k
    source lines */

    #ifdef __cplusplus
    extern "C"{
    #endif

    #include <rpc.h>
    #include <rpcndr.h>

    #ifdef _MIDL_USE_GUIDDEF_

    #ifndef INITGUID
    #define INITGUID
    #include <guiddef.h>
    #undef INITGUID
    #else
    #include <guiddef.h>
    #endif

    #define
    MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
    b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

    #else // !_MIDL_USE_GUIDDEF_

    #ifndef __IID_DEFINED__
    #define __IID_DEFINED__

    typedef struct _IID
    {
        unsigned long x;
        unsigned short s1;
        unsigned short s2;
        unsigned char c[8];
    } IID;

    #endif // __IID_DEFINED__

    #ifndef CLSID_DEFINED
    #define CLSID_DEFINED
    typedef IID CLSID;
    #endif // CLSID_DEFINED

    #define
    MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
    b7,b8) \
        const type name =
    {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

```

```

    #endif !_MIDL_USE_GUIDDEF_

    MIDL_DEFINE_GUID(IID,
    IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC
    0,0x4F,0xBF,0xE0,0x8B);

    #undef MIDL_DEFINE_GUID

    #ifdef __cplusplus
    }
    #endif

    #endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

    /* 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 6.00.0361
    */
    /* at Thu Mar 16 18:21:12 2006
    */
    /* 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)

    #pragma warning( disable: 4049 ) /* more than 64k
    source lines */

    #ifdef __cplusplus
    extern "C"{
    #endif

    #include <rpc.h>
    #include <rpcndr.h>

    #ifdef _MIDL_USE_GUIDDEF_

    #ifndef INITGUID
    #define INITGUID
    #include <guiddef.h>
    #undef INITGUID
    #else

```

```

    #include <guiddef.h>
    #endif

    #define
    MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
    b7,b8) \
    DEFINE_GUID(name,l,w1,w2,b1,b2,b3,b4,b5,b6,b7,b8)

    #else // !_MIDL_USE_GUIDDEF_

    #ifndef __IID_DEFINED__
    #define __IID_DEFINED__

    typedef struct _IID
    {
        unsigned long x;
        unsigned short s1;
        unsigned short s2;
        unsigned char c[8];
    } IID;

    #endif // __IID_DEFINED__

    #ifndef CLSID_DEFINED
    #define CLSID_DEFINED
    typedef IID CLSID;
    #endif // CLSID_DEFINED

    #define
    MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,b3,b4,b5,b6,
    b7,b8) \
        const type name =
    {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

    #endif !_MIDL_USE_GUIDDEF_

    MIDL_DEFINE_GUID(IID,
    IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC
    0,0x4F,0xBF,0xE0,0x8B);

    #undef MIDL_DEFINE_GUID

    #ifdef __cplusplus
    }
    #endif

    #endif /* defined(_M_IA64) || defined(_M_AMD64)*/



---



tpcc_com_ps_  
p.c



```

 /* this ALWAYS GENERATED file contains the proxy stub
 code */

```


```

```

/* File created by MIDL compiler version 6.00.0361
*/
/* at Thu Mar 16 18:21:12 2006
*/
/*
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)

#pragma warning( disable: 4049 ) /* more than 64k
source lines */
#if _MSC_VER >= 1200
#pragma warning(push)
#endif
#pragma warning( disable: 4100 ) /* unreferenced
arguments in x86 call */
#pragma warning( disable: 4211 ) /* redefine extent
to static */
#pragma warning( disable: 4232 ) /* dllimport
identity*/
#define USE_STUBLESS_PROXY

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

#include "rpcproxy.h"
#ifdef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of
<rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1023
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
short Pad;
unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{

```

```

short Pad;
unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,
0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(_RPC_WIN32_)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this
stub because it uses these features:
#error -Oif or -Oicf, [wire_marshal] or
[user_marshal] attribute.
#error However, your C/C++ compilation flags indicate
you intend to run this app on earlier systems.
#error This app will die there with the
RPC_X_WRONG_STUB_VERSION error.
#endif

static const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString =
{
0,
{
/* Procedure NewOrder */

FC_AUTO_HANDLE */
0x33, /*
0x6c, /*
Old Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ), /* 0 */
/* 6 */ NdrFcShort( 0x3 ), /* 3 */
/* 8 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 10 */ NdrFcShort( 0x0 ), /* 0 */
/* 12 */ NdrFcShort( 0x8 ), /* 8 */
/* 14 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */

```

```

0x3, /*
3 */ /* Parameter txn_in */

/* 16 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 18 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 20 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 22 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 24 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 26 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 28 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 30 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 32 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Payment */

/* 34 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ), /* 0 */
/* 40 */ NdrFcShort( 0x4 ), /* 4 */
/* 42 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 44 */ NdrFcShort( 0x0 ), /* 0 */
/* 46 */ NdrFcShort( 0x8 ), /* 8 */
/* 48 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */ /* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 52 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 54 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */

```

```

/* 58 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 60 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 64 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 66 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure Delivery */

/* 68 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ), /* 0 */
/* 74 */ NdrFcShort( 0x5 ), /* 5 */
/* 76 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 78 */ NdrFcShort( 0x0 ), /* 0 */
/* 80 */ NdrFcShort( 0x8 ), /* 8 */
/* 82 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 86 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 88 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 90 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 94 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 100 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure StockLevel */

/* 102 */ 0x33, /* FC_AUTO_HANDLE */

```

```

0x6c, /*
Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 122 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 128 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 134 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure OrderStatus */

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack
size/offset = 28 */
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */
/* 150 */ 0x7, /* Oi2 Flags: srv must
size, clt must size, has return, */
0x3, /*
3 */

/* Parameter txn_in */

/* 152 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */

```

```

/* 154 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 156 */ NdrFcShort( 0x3e2 ), /* Type
Offset=994 */

/* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=16 */
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack
size/offset = 20 */
/* 162 */ NdrFcShort( 0x3f4 ), /* Type
Offset=1012 */

/* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack
size/offset = 24 */
/* 168 */ 0x8, /* FC_LONG */
0x0, /*
0 */

/* Procedure CallSetComplete */

/* 170 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 172 */ NdrFcLong( 0x0 ), /* 0 */
/* 176 */ NdrFcShort( 0x8 ), /* 8 */
/* 178 */ NdrFcShort( 0x8 ), /* x86 Stack
size/offset = 8 */
/* 180 */ NdrFcShort( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x8 ), /* 8 */
/* 184 */ 0x4, /* Oi2 Flags: has
return, */
0x1, /*
1 */

/* Return value */

/* 186 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 188 */ NdrFcShort( 0x4 ), /* x86 Stack
size/offset = 4 */
/* 190 */ 0x8, /* FC_LONG */
0x0, /*
0 */

0x0
};

static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /*
0 */
/* 2 */

```

```

0x12, 0x0, /*
FC_UP */
/* 4 */ NdrFcShort( 0x3ca ), /* Offset=
970 (974) */
/* 6 */
FC_NON_ENCAPSULATED_UNION */
0x2b, /*
FC_ULONG */
0x9, /*
/*
/* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
0x0, /*
*/
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* 22 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 28 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* 34 */ NdrFcShort( 0x8001 ), /* Simple arm
type: FC_BYTE */
/* 36 */ NdrFcLong( 0x2 ), /* 2 */
/* 40 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x800a ), /* Simple arm
type: FC_FLOAT */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
/* 54 */ NdrFcLong( 0xb ), /* 11 */
/* 58 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 60 */ NdrFcLong( 0xa ), /* 10 */
/* 64 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 66 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /* Offset= 232 (302) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 76 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
/* 78 */ NdrFcLong( 0x8 ), /* 8 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0xd ), /* 13 */
/* 88 */ NdrFcShort( 0xf4 ), /* Offset= 244 (332) */
/* 90 */ NdrFcLong( 0x9 ), /* 9 */
/* 94 */ NdrFcShort( 0x100 ), /* Offset=
256 (350) */
/* 96 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 100 */ NdrFcShort( 0x10c ), /* Offset=
268 (368) */
/* 102 */ NdrFcLong( 0x24 ), /* 36 */
/* 106 */ NdrFcShort( 0x31a ), /* Offset=
794 (900) */
/* 108 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 112 */ NdrFcShort( 0x314 ), /* Offset=
788 (900) */
/* 114 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 118 */ NdrFcShort( 0x312 ), /* Offset=
786 (904) */
/* 120 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 124 */ NdrFcShort( 0x310 ), /* Offset=
784 (908) */
/* 126 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 130 */ NdrFcShort( 0x30e ), /* Offset=
782 (912) */
/* 132 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 136 */ NdrFcShort( 0x30c ), /* Offset=
780 (916) */
/* 138 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 142 */ NdrFcShort( 0x30a ), /* Offset=
778 (920) */
/* 144 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 148 */ NdrFcShort( 0x308 ), /* Offset=
776 (924) */
/* 150 */ NdrFcLong( 0x400b ), /* 16395 */
/* 154 */ NdrFcShort( 0x2f2 ), /* Offset=
754 (908) */
/* 156 */ NdrFcLong( 0x400a ), /* 16394 */
/* 160 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (912) */
/* 162 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 166 */ NdrFcShort( 0x2fa ), /* Offset=
762 (928) */
/* 168 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 172 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (924) */
/* 174 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 178 */ NdrFcShort( 0x2f2 ), /* Offset=
754 (932) */
/* 180 */ NdrFcLong( 0x400d ), /* 16397 */
/* 184 */ NdrFcShort( 0x2f0 ), /* Offset=
752 (936) */
/* 186 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 190 */ NdrFcShort( 0x2ee ), /* Offset=
750 (940) */
/* 192 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 196 */ NdrFcShort( 0x2ec ), /* Offset=
748 (944) */
/* 198 */ NdrFcLong( 0x400c ), /* 16396 */
/* 202 */ NdrFcShort( 0x2ea ), /* Offset=
746 (948) */
/* 204 */ NdrFcLong( 0x10 ), /* 16 */
/* 208 */ NdrFcShort( 0x8002 ), /* Simple arm
type: FC_CHAR */
/* 210 */ NdrFcLong( 0x12 ), /* 18 */
/* 214 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
/* 216 */ NdrFcLong( 0x13 ), /* 19 */
/* 220 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 222 */ NdrFcLong( 0x15 ), /* 21 */
/* 226 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 228 */ NdrFcLong( 0x16 ), /* 22 */
/* 232 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 234 */ NdrFcLong( 0x17 ), /* 23 */
/* 238 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 240 */ NdrFcLong( 0xe ), /* 14 */
/* 244 */ NdrFcShort( 0x2c8 ), /* Offset=
712 (956) */
/* 246 */ NdrFcLong( 0x400e ), /* 16398 */
/* 250 */ NdrFcShort( 0x2cc ), /* Offset=
716 (966) */
/* 252 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 256 */ NdrFcShort( 0x2ca ), /* Offset=
714 (970) */
/* 258 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 262 */ NdrFcShort( 0x286 ), /* Offset=
646 (908) */
/* 264 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 268 */ NdrFcShort( 0x284 ), /* Offset=
644 (912) */
/* 270 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 274 */ NdrFcShort( 0x282 ), /* Offset=
642 (916) */
/* 276 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 280 */ NdrFcShort( 0x278 ), /* Offset=
632 (912) */
/* 282 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 286 */ NdrFcShort( 0x272 ), /* Offset=
626 (912) */
/* 288 */ NdrFcLong( 0x0 ), /* 0 */
/* 292 */ NdrFcShort( 0x0 ), /* Offset= 0 (292) */
/* 294 */ NdrFcLong( 0x1 ), /* 1 */
/* 298 */ NdrFcShort( 0x0 ), /* Offset= 0 (298) */
/* 300 */ NdrFcShort( 0xffff ), /* Offset= -1
(299) */
/* 302 */
FC_STRUCT */
0x15, /*
7 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 308 */
0x12, 0x0, /*
FC_UP */
/* 310 */ NdrFcShort( 0xc ), /* Offset= 12 (322) */
/* 312 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 314 */ NdrFcShort( 0x2 ), /* 2 */
/* 316 */ 0x9, /* Corr desc: FC_ULONG
*/
0x0, /*
*/
/* 318 */ NdrFcShort( 0xffffc ), /* -4 */
/* 320 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 322 */
0x17, /*
FC_CSTRUCT */
0x3, /*
3 */
/* 324 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 326 */ NdrFcShort( 0xffff2 ), /* Offset= -
14 (312) */
/* 328 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 330 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 332 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 334 */ NdrFcLong( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ NdrFcShort( 0x0 ), /* 0 */
/* 342 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 344 */ 0x0, /* 0 */
0x0, /*
0 */
/* 346 */ 0x0, /* 0 */
0x0, /*
0 */
/* 348 */ 0x0, /* 0 */
0x46, /*
70 */
/* 350 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 352 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 356 */ NdrFcShort( 0x0 ), /* 0 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* 360 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 362 */ 0x0, /* 0 */
0x0, /*
0 */
/* 364 */ 0x0, /* 0 */
0x0, /*
0 */
/* 366 */ 0x0, /* 0 */
0x46, /*
70 */
/* 368 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 370 */ NdrFcShort( 0x2 ), /* Offset= 2 (372) */
/* 372 */
0x12, 0x0, /*
FC_UP */
/* 374 */ NdrFcShort( 0x1fc ), /* Offset=
508 (882) */
/* 376 */
0x2a, /*
FC_ENCAPSULATED_UNION */
0x49, /*
73 */
/* 378 */ NdrFcShort( 0x18 ), /* 24 */

```

```

/* 380 */ NdrFcShort( 0xa ), /* 10 */
/* 382 */ NdrFcLong( 0x8 ), /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /* Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ), /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /* Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ), /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /* Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ), /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /* Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ), /* 36 */
/* 410 */ NdrFcShort( 0x114 ), /* Offset=
276 (686) */
/* 412 */ NdrFcLong( 0x800d ), /* 32781 */
/* 416 */ NdrFcShort( 0x130 ), /* Offset=
304 (720) */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /* Offset=
328 (750) */
/* 424 */ NdrFcLong( 0x2 ), /* 2 */
/* 428 */ NdrFcShort( 0x160 ), /* Offset=
352 (780) */
/* 430 */ NdrFcLong( 0x3 ), /* 3 */
/* 434 */ NdrFcShort( 0x178 ), /* Offset=
376 (810) */
/* 436 */ NdrFcLong( 0x14 ), /* 20 */
/* 440 */ NdrFcShort( 0x190 ), /* Offset=
400 (840) */
/* 442 */ NdrFcShort( 0xfffff ), /* Offset= -1
(441) */
/* 444 */
0x1b, /*
FC_CARRAY */
0x3, /*
3 */
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 454 */
0x48, /*
FC_VARIABLE_REPEAT */
0x49, /*
FC_FIXED_OFFSET */
/* 456 */ NdrFcShort( 0x4 ), /* 4 */
/* 458 */ NdrFcShort( 0x0 ), /* 0 */
/* 460 */ NdrFcShort( 0x1 ), /* 1 */
/* 462 */ NdrFcShort( 0x0 ), /* 0 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x12, 0x0, /* FC_UP */
/* 468 */ NdrFcShort( 0xff6e ), /* Offset= -
146 (322) */
/* 470 */
0x5b, /*
FC_END */

```

```

0x8, /*
FC_LONG */
/* 472 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 474 */
0x16, /*
FC_PSTRUCT */
0x3, /*
3 */
/* 476 */ NdrFcShort( 0x8 ), /* 8 */
/* 478 */
0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 480 */
0x46, /*
FC_NO_REPEAT */
0x5c, /*
FC_PAD */
/* 482 */ NdrFcShort( 0x4 ), /* 4 */
/* 484 */ NdrFcShort( 0x4 ), /* 4 */
/* 486 */ 0x11, 0x0, /* FC_RP */
/* 488 */ NdrFcShort( 0xffd4 ), /* Offset= -
44 (444) */
/* 490 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 492 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 494 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 496 */ NdrFcShort( 0x0 ), /* 0 */
/* 498 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 500 */ NdrFcShort( 0x0 ), /* 0 */
/* 502 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 506 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 */
/* 508 */ NdrFcShort( 0xff50 ), /* Offset= -
176 (332) */
/* 510 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 512 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 514 */ NdrFcShort( 0x8 ), /* 8 */
/* 516 */ NdrFcShort( 0x0 ), /* 0 */

```



```

/* 518 */ NdrFcShort( 0x6 ), /* Offset= 6 (524) */
/* 520 */ 0x8, /* FC_LONG */
/* 522 */ 0x5c, /* FC_PAD */
/* 524 */ 0x5b, /*
FC_POINTER */
/* 524 */ 0x11, 0x0, /*
FC_END */
/* 524 */
FC_RP */
/* 526 */ NdrFcShort( 0xffe0 ), /* Offset= -
32 (494) */
/* 528 */
/* 528 */ 0x21, /*
FC_BOGUS_ARRAY */
/* 528 */ 0x3, /*
3 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/* 532 */ 0x0, /*
*/
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
/* 540 */ 0x0, /*
0 */
/* 542 */ NdrFcShort( 0xff40 ), /* Offset= -
192 (350) */
/* 544 */ 0x5c, /* FC_PAD */
/* 544 */ 0x5b, /*
FC_END */
/* 546 */
/* 546 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 546 */ 0x3, /*
3 */
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8, /* FC_LONG */
/* 554 */ 0x36, /*
FC_POINTER */
/* 556 */ 0x5c, /* FC_PAD */
/* 556 */ 0x5b, /*
FC_END */
/* 558 */
/* 558 */ 0x11, 0x0, /*
FC_RP */
/* 560 */ NdrFcShort( 0xffe0 ), /* Offset= -
32 (528) */
/* 562 */
/* 562 */ 0x1b, /*
FC_CARRAY */
/* 562 */ 0x3, /*
3 */
/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/* 566 */ 0x0, /*
*/
/* 568 */ NdrFcShort( 0x0 ), /* 0 */

```

```

/* 570 */
FC_PP */
/* 570 */ 0x4b, /*
*/
FC_PAD */
/* 572 */ 0x5c, /*
*/
FC_VARIABLE_REPEAT */
/* 572 */ 0x48, /*
*/
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 */
/* 588 */ 0x5b, /*
FC_END */
/* 588 */ 0x8, /*
FC_LONG */
/* 590 */ 0x5c, /* FC_PAD */
/* 590 */ 0x5b, /*
FC_END */
/* 592 */
/* 592 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 592 */ 0x3, /*
3 */
/* 594 */ NdrFcShort( 0x8 ), /* 8 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x6 ), /* Offset= 6 (604) */
/* 600 */ 0x8, /* FC_LONG */
/* 600 */ 0x36, /*
FC_POINTER */
/* 602 */ 0x5c, /* FC_PAD */
/* 602 */ 0x5b, /*
FC_END */
/* 604 */
/* 604 */ 0x11, 0x0, /*
FC_RP */
/* 606 */ NdrFcShort( 0xffd4 ), /* Offset= -
44 (562) */
/* 608 */
/* 608 */ 0x2f, /*
FC_IP */
/* 608 */ 0x5a, /*
FC_CONSTANT_IID */
/* 610 */ NdrFcLong( 0x2f ), /* 47 */
/* 614 */ NdrFcShort( 0x0 ), /* 0 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ 0xc0, /* 192 */
/* 618 */ 0x0, /*
0 */
/* 620 */ 0x0, /* 0 */
/* 620 */ 0x0, /*
0 */
/* 622 */ 0x0, /* 0 */
/* 622 */ 0x0, /*
0 */
0 */

```

```

/* 624 */ 0x0, /* 0 */
/* 624 */ 0x46, /*
70 */
/* 626 */
FC_CARRAY */
/* 626 */ 0x1b, /*
0 */
/* 628 */ NdrFcShort( 0x1 ), /* 1 */
/* 630 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/* 630 */ 0x0, /*
*/
/* 632 */ NdrFcShort( 0x4 ), /* 4 */
/* 634 */ 0x1, /* FC_BYTE */
/* 634 */ 0x5b, /*
FC_END */
/* 636 */
/* 636 */ 0x1a, /*
FC_BOGUS_STRUCT */
/* 636 */ 0x3, /*
3 */
/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0xa ), /* Offset= 10 (652) */
/* 644 */ 0x8, /* FC_LONG */
/* 644 */ 0x8, /*
FC_LONG */
/* 646 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
/* 646 */ 0x0, /*
0 */
/* 648 */ NdrFcShort( 0xffd8 ), /* Offset= -
40 (608) */
/* 650 */ 0x36, /* FC_POINTER */
/* 650 */ 0x5b, /*
FC_END */
/* 652 */
/* 652 */ 0x12, 0x0, /*
FC_UP */
/* 654 */ NdrFcShort( 0xffe4 ), /* Offset= -
28 (626) */
/* 656 */
/* 656 */ 0x1b, /*
FC_CARRAY */
/* 656 */ 0x3, /*
3 */
/* 658 */ NdrFcShort( 0x4 ), /* 4 */
/* 660 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
/* 660 */ 0x0, /*
*/
/* 662 */ NdrFcShort( 0x0 ), /* 0 */
/* 664 */
/* 664 */ 0x4b, /*
FC_PP */
/* 664 */ 0x5c, /*
FC_PAD */
/* 666 */
/* 666 */ 0x48, /*
FC_VARIABLE_REPEAT */
/* 666 */ 0x49, /*
FC_FIXED_OFFSET */

```

```

/* 668 */ NdrFcShort( 0x4 ), /* 4 */
/* 670 */ NdrFcShort( 0x0 ), /* 0 */
/* 672 */ NdrFcShort( 0x1 ), /* 1 */
/* 674 */ NdrFcShort( 0x0 ), /* 0 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ 0x12, 0x0, /* FC_UP */
/* 680 */ NdrFcShort( 0xffd4 ), /* Offset= -
44 (636) */
/* 682 */
FC_END /*
0x5b, /*
FC_LONG /*
0x8, /*
/* 684 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 686 */
FC_BOGUS_STRUCT /*
0x3, /*
3 /*
/* 688 */ NdrFcShort( 0x8 ), /* 8 */
/* 690 */ NdrFcShort( 0x0 ), /* 0 */
/* 692 */ NdrFcShort( 0x6 ), /* Offset= 6 (698) */
/* 694 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER /*
/* 696 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 698 */
0x11, 0x0, /*
FC_RP /*
/* 700 */ NdrFcShort( 0xffd4 ), /* Offset= -
44 (656) */
/* 702 */
0x1d, /*
FC_SMFARRAY /*
0x0, /*
0 /*
/* 704 */ NdrFcShort( 0x8 ), /* 8 */
/* 706 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END /*
/* 708 */
0x15, /*
FC_STRUCT /*
0x3, /*
3 /*
/* 710 */ NdrFcShort( 0x10 ), /* 16 */
/* 712 */ 0x8, /* FC_LONG */
0x6, /*
FC_SHORT /*
/* 714 */ 0x6, /* FC_SHORT */
0x4c, /*
FC_EMBEDDED_COMPLEX /*
/* 716 */ 0x0, /* 0 */
NdrFcShort( 0xffff1 ),
/* Offset= -15 (702) */
0x5b, /*
FC_END /*
/* 720 */

```

```

FC_BOGUS_STRUCT /*
0x1a, /*
0x3, /*
3 /*
/* 722 */ NdrFcShort( 0x18 ), /* 24 */
/* 724 */ NdrFcShort( 0x0 ), /* 0 */
/* 726 */ NdrFcShort( 0xa ), /* Offset= 10 (736) */
/* 728 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER /*
/* 730 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /*
0 /*
/* 732 */ NdrFcShort( 0xffe8 ), /* Offset= -
24 (708) */
/* 734 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END /*
/* 736 */
0x11, 0x0, /*
FC_RP /*
/* 738 */ NdrFcShort( 0xff0c ), /* Offset= -
244 (494) */
/* 740 */
0x1b, /*
FC_CARRAY /*
0x0, /*
0 /*
/* 742 */ NdrFcShort( 0x1 ), /* 1 */
/* 744 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/*
/* 746 */ NdrFcShort( 0x0 ), /* 0 */
/* 748 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END /*
/* 750 */
0x16, /*
FC_PSTRUCT /*
0x3, /*
3 /*
/* 752 */ NdrFcShort( 0x8 ), /* 8 */
/* 754 */
0x4b, /*
FC_PP /*
0x5c, /*
FC_PAD /*
/* 756 */
0x46, /*
FC_NO_REPEAT /*
0x5c, /*
FC_PAD /*
/* 758 */ NdrFcShort( 0x4 ), /* 4 */
/* 760 */ NdrFcShort( 0x4 ), /* 4 */
/* 762 */ 0x12, 0x0, /* FC_UP */
/* 764 */ NdrFcShort( 0xffe8 ), /* Offset= -
24 (740) */
/* 766 */
0x5b, /*
FC_END /*

```

```

0x8, /*
FC_LONG /*
/* 768 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*
/* 770 */
FC_CARRAY /*
0x1b, /*
0x1, /*
1 /*
/* 772 */ NdrFcShort( 0x2 ), /* 2 */
/* 774 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/*
/* 776 */ NdrFcShort( 0x0 ), /* 0 */
/* 778 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END /*
/* 780 */
0x16, /*
FC_PSTRUCT /*
0x3, /*
3 /*
/* 782 */ NdrFcShort( 0x8 ), /* 8 */
/* 784 */
0x4b, /*
FC_PP /*
0x5c, /*
FC_PAD /*
/* 786 */
0x46, /*
FC_NO_REPEAT /*
0x5c, /*
FC_PAD /*
/* 788 */ NdrFcShort( 0x4 ), /* 4 */
/* 790 */ NdrFcShort( 0x4 ), /* 4 */
/* 792 */ 0x12, 0x0, /* FC_UP */
/* 794 */ NdrFcShort( 0xffe8 ), /* Offset= -
24 (770) */
/* 796 */
0x5b, /*
FC_END /*
0x8, /*
FC_LONG /*
/* 798 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*
/* 800 */
FC_CARRAY /*
0x3, /*
3 /*
/* 802 */ NdrFcShort( 0x4 ), /* 4 */
/* 804 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
/*
/* 806 */ NdrFcShort( 0x0 ), /* 0 */
/* 808 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END /*

```

```

/* 810 */
FC_PSTRUCT */
3 */
/* 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( 0xffe8 ), /* Offset= -
24 (800) */
/* 826 */
FC_END */
FC_LONG */
/* 828 */ 0x8, /* FC_LONG */
FC_END */
/* 830 */
FC_CARRAY */
7 */
/* 832 */ NdrFcShort( 0x8 ), /* 8 */
/* 834 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
*/
/* 836 */ NdrFcShort( 0x0 ), /* 0 */
/* 838 */ 0xb, /* FC_HYPER */
FC_END */
/* 840 */
FC_PSTRUCT */
3 */
/* 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( 0xffe8 ), /* Offset= -
24 (830) */
/* 856 */
FC_END */
FC_LONG */
/* 858 */ 0x8, /* FC_LONG */
FC_END */
/* 860 */
FC_STRUCT */
3 */
/* 862 */ NdrFcShort( 0x8 ), /* 8 */
/* 864 */ 0x8, /* FC_LONG */
FC_LONG */
/* 866 */ 0x5c, /* FC_PAD */
FC_END */
/* 868 */
FC_CARRAY */
3 */
/* 870 */ NdrFcShort( 0x8 ), /* 8 */
/* 872 */ 0x7, /* Corr desc: FC_USHORT
*/
*/
/* 874 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 876 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0 */
/* 878 */ NdrFcShort( 0xffee ), /* Offset= -
18 (860) */
/* 880 */ 0x5c, /* FC_PAD */
FC_END */
/* 882 */
FC_BOGUS_STRUCT */
3 */
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xffee ), /* Offset= -
18 (868) */
/* 888 */ NdrFcShort( 0x0 ), /* Offset= 0 (888) */
/* 890 */ 0x6, /* FC_SHORT */
FC_SHORT */
/* 892 */ 0x8, /* FC_LONG */
FC_LONG */
/* 894 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0 */

```

```

/* 896 */ NdrFcShort( 0xdf8 ), /* Offset= -
520 (376) */
/* 898 */ 0x5c, /* FC_PAD */
FC_END */
/* 900 */
FC_UP */
/* 902 */ NdrFcShort( 0xfef6 ), /* Offset= -
266 (636) */
/* 904 */
FC_UP [simple_pointer] */
/* 906 */ 0x1, /* FC_BYTE */
FC_PAD */
/* 908 */
FC_UP [simple_pointer] */
/* 910 */ 0x6, /* FC_SHORT */
FC_PAD */
/* 912 */
FC_UP [simple_pointer] */
/* 914 */ 0x8, /* FC_LONG */
FC_PAD */
/* 916 */
FC_UP [simple_pointer] */
/* 918 */ 0xb, /* FC_HYPER */
FC_PAD */
/* 920 */
FC_UP [simple_pointer] */
/* 922 */ 0xa, /* FC_FLOAT */
FC_PAD */
/* 924 */
FC_UP [simple_pointer] */
/* 926 */ 0xc, /* FC_DOUBLE */
FC_PAD */
/* 928 */
FC_UP */
/* 930 */ NdrFcShort( 0xfd8c ), /* Offset= -
628 (302) */
/* 932 */
FC_UP [pointer_deref] */
/* 934 */ NdrFcShort( 0xfd8e ), /* Offset= -
626 (308) */
/* 936 */
FC_UP [pointer_deref] */
/* 938 */ NdrFcShort( 0xfda2 ), /* Offset= -
606 (332) */
/* 940 */

```

```

                                0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 942 */ NdrFcShort( 0xfdb0 ), /* Offset= -
592 (350) */
/* 944 */
                                0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 946 */ NdrFcShort( 0xfdb0 ), /* Offset= -
578 (368) */
/* 948 */
                                0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 950 */ NdrFcShort( 0x2 ), /* Offset= 2 (952) */
/* 952 */
                                0x12, 0x0, /*
FC_UP */
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
FC_STRUCT */
                                0x7, /*
7 */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /* FC_SHORT */
                                0x1, /*
FC_BYTE */
/* 962 */ 0x1, /* FC_BYTE */
                                0x8, /*
FC_LONG */
/* 964 */ 0xb, /* FC_HYPER */
                                0x5b, /*
FC_END */
/* 966 */
                                0x12, 0x0, /*
FC_UP */
/* 968 */ NdrFcShort( 0xffff4 ), /* Offset= -
12 (956) */
/* 970 */
                                0x12, 0x8, /*
FC_UP [simple_pointer] */
/* 972 */ 0x2, /* FC_CHAR */
                                0x5c, /*
FC_PAD */
/* 974 */
                                0x1a, /*
FC_BOGUS_STRUCT */
                                0x7, /*
7 */
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ), /* 0 */
/* 980 */ NdrFcShort( 0x0 ), /* Offset= 0 (980) */
/* 982 */ 0x8, /* FC_LONG */
                                0x8, /*
FC_LONG */
/* 984 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 986 */ 0x6, /* FC_SHORT */
                                0x6, /*
FC_SHORT */
/* 988 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/

```

```

                                0x0, /*
0 */
/* 990 */ NdrFcShort( 0xfc28 ), /* Offset= -
984 (6) */
/* 992 */ 0x5c, /* FC_PAD */
                                0x5b, /*
FC_END */
/* 994 */ 0xb4, /* FC_USER_MARSHAL */
                                0x83, /*
131 */
/* 996 */ NdrFcShort( 0x0 ), /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
/* 1002 */ NdrFcShort( 0xfc18 ), /*
Offset= -1000 (2) */
/* 1004 */
                                0x11, 0x4, /*
FC_RP [allocated_on_stack] */
/* 1006 */ NdrFcShort( 0x6 ), /* Offset= 6
(1012) */
/* 1008 */
                                0x13, 0x0, /*
FC_OP */
/* 1010 */ NdrFcShort( 0xffdc ), /*
Offset= -36 (974) */
/* 1012 */ 0xb4, /*
FC_USER_MARSHAL */
                                0x83, /*
131 */
/* 1014 */ NdrFcShort( 0x0 ), /* 0 */
/* 1016 */ NdrFcShort( 0x10 ), /* 16 */
/* 1018 */ NdrFcShort( 0x0 ), /* 0 */
/* 1020 */ NdrFcShort( 0xffff4 ), /*
Offset= -12 (1008) */
                                0x0
}
};

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000,
ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,

```

```

GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable[] =
{
    0,
    34,
    68,
    102,
    136,
    170
};

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo
=
{
    &Object_StubDesc,
    _MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    _MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0
};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy ,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete
*/
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,

```



```

#if !defined( _RPC_WIN64 )
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString =
{
    0,
    {
        /* Procedure NewOrder */
        0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */
        /* 2 */ NdrFcLong( 0x0 ), /* 0 */
        /* 6 */ NdrFcShort( 0x3 ), /* 3 */
        /* 8 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
        /* 10 */ NdrFcShort( 0x0 ), /* 0 */
        /* 12 */ NdrFcShort( 0x8 ), /* 8 */
        /* 14 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
        0x3, /* 3 */
        /* 16 */ 0xa, /* 10 */
        0x7, /* Ext Flags: new corr desc, clt corr check, srv corr
check, */
        /* 18 */ NdrFcShort( 0x20 ), /* 32 */
        /* 20 */ NdrFcShort( 0x20 ), /* 32 */
        /* 22 */ NdrFcShort( 0x0 ), /* 0 */
        /* 24 */ NdrFcShort( 0x0 ), /* 0 */

        /* Parameter txn_in */
        /* 26 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
        /* 28 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
        /* 30 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

        /* Parameter txn_out */
        /* 32 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
        /* 34 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
        /* 36 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

        /* Return value */
        /* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
        /* 40 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
        /* 42 */ 0x8, /* FC_LONG */
        0x0, /* 0 */
    }
}

```

```

/* Procedure Payment */
/* 44 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
/* 52 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
0x3, /* 3 */
/* 60 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */
/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 72 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 74 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */
/* 76 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 78 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 80 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */
/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 84 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 86 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */
/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
/* 96 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */

```

```

/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
0x3, /* 3 */
/* 104 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */
/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 116 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 118 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

/* Parameter txn_out */
/* 120 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 122 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 124 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

/* Return value */
/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 128 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 130 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */
/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
/* 140 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
0x3, /* 3 */
/* 148 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt corr check, srv corr
check, */

```

```

/* 150 */ NdrFcShort( 0x20 ), /* 32 */
/* 152 */ NdrFcShort( 0x20 ), /* 32 */
/* 154 */ NdrFcShort( 0x0 ), /* 0 */
/* 156 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 158 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */
/* 160 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 162 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

    /* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 166 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 168 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

    /* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 172 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 174 */ 0x8, /* FC_LONG */
0x0, /*
0 */

    /* Procedure OrderStatus */

/* 176 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 178 */ NdrFcLong( 0x0 ), /* 0 */
/* 182 */ NdrFcShort( 0x7 ), /* 7 */
/* 184 */ NdrFcShort( 0x30 ), /* ia64 Stack
size/offset = 48 */
/* 186 */ NdrFcShort( 0x0 ), /* 0 */
/* 188 */ NdrFcShort( 0x8 ), /* 8 */
/* 190 */ 0x47, /* Oi2 Flags: srv must
size, clt must size, has return, has ext, */
0x3, /*
3 */
/* 192 */ 0xa, /* 10 */
0x7, /*
Ext Flags: new corr desc, clt corr check, srv corr
check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ), /* 0 */
/* 200 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags: must size,
must free, in, by val, */

```

```

/* 204 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 206 */ NdrFcShort( 0x3ce ), /* Type
Offset=974 */

    /* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ), /* Flags:
must size, must free, out, simple ref, srv alloc
size=24 */
/* 210 */ NdrFcShort( 0x20 ), /* ia64 Stack
size/offset = 32 */
/* 212 */ NdrFcShort( 0x3e0 ), /* Type
Offset=992 */

    /* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 216 */ NdrFcShort( 0x28 ), /* ia64 Stack
size/offset = 40 */
/* 218 */ 0x8, /* FC_LONG */
0x0, /*
0 */

    /* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /*
Old Flags: object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ), /* 0 */
/* 226 */ NdrFcShort( 0x8 ), /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64 Stack
size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ), /* 0 */
/* 232 */ NdrFcShort( 0x8 ), /* 8 */
/* 234 */ 0x44, /* Oi2 Flags: has
return, has ext, */
0x1, /*
1 */
/* 236 */ 0xa, /* 10 */
0x1, /*
Ext Flags: new corr desc, */
/* 238 */ NdrFcShort( 0x0 ), /* 0 */
/* 240 */ NdrFcShort( 0x0 ), /* 0 */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */

    /* Return value */

/* 246 */ NdrFcShort( 0x70 ), /* Flags: out, return,
base type, */
/* 248 */ NdrFcShort( 0x8 ), /* ia64 Stack
size/offset = 8 */
/* 250 */ 0x8, /* FC_LONG */
0x0, /*
0 */

    }
};

```

```

static const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /*
0 */
        /* 2 */
        0x12, 0x0, /*
FC_UP */
        /* 4 */ NdrFcShort( 0x3b6 ), /* Offset=
950 (954) */
        /* 6 */
        0x2b, /*
FC_NON_ENCAPSULATED_UNION */
        0x9, /*
FC_ULONG */
        /* 8 */ 0x7, /* Corr desc: FC_USHORT
*/
        0x0, /*
*/
        /* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
        /* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
        /* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
        /* 16 */ NdrFcShort( 0x10 ), /* 16 */
        /* 18 */ NdrFcShort( 0x2f ), /* 47 */
        /* 20 */ NdrFcLong( 0x14 ), /* 20 */
        /* 24 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
        /* 26 */ NdrFcLong( 0x3 ), /* 3 */
        /* 30 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
        /* 32 */ NdrFcLong( 0x11 ), /* 17 */
        /* 36 */ NdrFcShort( 0x8001 ), /* Simple arm
type: FC_BYTE */
        /* 38 */ NdrFcLong( 0x2 ), /* 2 */
        /* 42 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
        /* 44 */ NdrFcLong( 0x4 ), /* 4 */
        /* 48 */ NdrFcShort( 0x800a ), /* Simple arm
type: FC_FLOAT */
        /* 50 */ NdrFcLong( 0x5 ), /* 5 */
        /* 54 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
        /* 56 */ NdrFcLong( 0xb ), /* 11 */
        /* 60 */ NdrFcShort( 0x8006 ), /* Simple arm
type: FC_SHORT */
        /* 62 */ NdrFcLong( 0xa ), /* 10 */
        /* 66 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
        /* 68 */ NdrFcLong( 0x6 ), /* 6 */
        /* 72 */ NdrFcShort( 0xe8 ), /* Offset= 232 (304) */
        /* 74 */ NdrFcLong( 0x7 ), /* 7 */
        /* 78 */ NdrFcShort( 0x800c ), /* Simple arm
type: FC_DOUBLE */
        /* 80 */ NdrFcLong( 0x8 ), /* 8 */
        /* 84 */ NdrFcShort( 0xe2 ), /* Offset= 226 (310) */
        /* 86 */ NdrFcLong( 0xd ), /* 13 */
        /* 90 */ NdrFcShort( 0xf6 ), /* Offset= 246 (336) */
        /* 92 */ NdrFcLong( 0x9 ), /* 9 */
        /* 96 */ NdrFcShort( 0x102 ), /* Offset=
258 (354) */
    }
};

```

```

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

```

```

/* 224 */ NdrFcLong( 0x15 ), /* 21 */
/* 228 */ NdrFcShort( 0x800b ), /* Simple arm
type: FC_HYPER */
/* 230 */ NdrFcLong( 0x16 ), /* 22 */
/* 234 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 236 */ NdrFcLong( 0x17 ), /* 23 */
/* 240 */ NdrFcShort( 0x8008 ), /* Simple arm
type: FC_LONG */
/* 242 */ NdrFcLong( 0xe ), /* 14 */
/* 246 */ NdrFcShort( 0x2b2 ), /* Offset=
690 (936) */
/* 248 */ NdrFcLong( 0x400e ), /* 16398 */
/* 252 */ NdrFcShort( 0x2b6 ), /* Offset=
694 (946) */
/* 254 */ NdrFcLong( 0x4010 ), /* 16400 */
/* 258 */ NdrFcShort( 0x2b4 ), /* Offset=
692 (950) */
/* 260 */ NdrFcLong( 0x4012 ), /* 16402 */
/* 264 */ NdrFcShort( 0x270 ), /* Offset=
624 (888) */
/* 266 */ NdrFcLong( 0x4013 ), /* 16403 */
/* 270 */ NdrFcShort( 0x26e ), /* Offset=
622 (892) */
/* 272 */ NdrFcLong( 0x4015 ), /* 16405 */
/* 276 */ NdrFcShort( 0x26c ), /* Offset=
620 (896) */
/* 278 */ NdrFcLong( 0x4016 ), /* 16406 */
/* 282 */ NdrFcShort( 0x262 ), /* Offset=
610 (892) */
/* 284 */ NdrFcLong( 0x4017 ), /* 16407 */
/* 288 */ NdrFcShort( 0x25c ), /* Offset=
604 (892) */
/* 290 */ NdrFcLong( 0x0 ), /* 0 */
/* 294 */ NdrFcShort( 0x0 ), /* Offset= 0 (294) */
/* 296 */ NdrFcLong( 0x1 ), /* 1 */
/* 300 */ NdrFcShort( 0x0 ), /* Offset= 0 (300) */
/* 302 */ NdrFcShort( 0xffff ), /* Offset= -1
(301) */
/* 304 */
FC_STRUCT */
0x15, /*
0x7, /*
7 */
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 310 */
0x12, 0x0, /*
FC_UP */
/* 312 */ NdrFcShort( 0xe ), /* Offset= 14 (312) */
/* 314 */
0x1b, /*
FC_CARRAY */
0x1, /*
1 */
/* 316 */ NdrFcShort( 0x2 ), /* 2 */
/* 318 */ 0x9, /* Corr desc: FC_ULONG
*/
0x0, /*
*/
/* 320 */ NdrFcShort( 0xfffc ), /* -4 */

```

```

/* 322 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 324 */ 0x6, /* FC_SHORT */
0x5b, /*
FC_END */
/* 326 */
FC_CSTRUCT */
0x17, /*
0x3, /*
3 */
/* 328 */ NdrFcShort( 0x8 ), /* 8 */
/* 330 */ NdrFcShort( 0xffff0 ), /* Offset= -
16 (314) */
/* 332 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 334 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 336 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x0 ), /* 0 */
/* 344 */ NdrFcShort( 0x0 ), /* 0 */
/* 346 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 348 */ 0x0, /* 0 */
0x0, /*
0 */
/* 350 */ 0x0, /* 0 */
0x0, /*
0 */
/* 352 */ 0x0, /* 0 */
0x46, /*
70 */
/* 354 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 356 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 360 */ NdrFcShort( 0x0 ), /* 0 */
/* 362 */ NdrFcShort( 0x0 ), /* 0 */
/* 364 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 366 */ 0x0, /* 0 */
0x0, /*
0 */
/* 368 */ 0x0, /* 0 */
0x0, /*
0 */
/* 370 */ 0x0, /* 0 */
0x46, /*
70 */
/* 372 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 374 */ NdrFcShort( 0x2 ), /* Offset= 2 (376) */

```



```

/* 376 */
FC_UP */
/* 378 */ NdrFcShort( 0x1e4 ), /* Offset=
484 (862) */
/* 380 */
FC_ENCAPSULATED_UNION */
0x2a, /*
0x89, /*
137 */
/* 382 */ NdrFcShort( 0x20 ), /* 32 */
/* 384 */ NdrFcShort( 0xa ), /* 10 */
/* 386 */ NdrFcLong( 0x8 ), /* 8 */
/* 390 */ NdrFcShort( 0x50 ), /* Offset= 80 (470) */
/* 392 */ NdrFcLong( 0xd ), /* 13 */
/* 396 */ NdrFcShort( 0x70 ), /* Offset= 112 (508) */
/* 398 */ NdrFcLong( 0x9 ), /* 9 */
/* 402 */ NdrFcShort( 0x90 ), /* Offset= 144 (546) */
/* 404 */ NdrFcLong( 0xc ), /* 12 */
/* 408 */ NdrFcShort( 0xb0 ), /* Offset= 176 (584) */
/* 410 */ NdrFcLong( 0x24 ), /* 36 */
/* 414 */ NdrFcShort( 0x102 ), /* Offset=
258 (672) */
/* 416 */ NdrFcLong( 0x800d ), /* 32781 */
/* 420 */ NdrFcShort( 0x11e ), /* Offset=
286 (706) */
/* 422 */ NdrFcLong( 0x10 ), /* 16 */
/* 426 */ NdrFcShort( 0x138 ), /* Offset=
312 (738) */
/* 428 */ NdrFcLong( 0x2 ), /* 2 */
/* 432 */ NdrFcShort( 0x14e ), /* Offset=
334 (766) */
/* 434 */ NdrFcLong( 0x3 ), /* 3 */
/* 438 */ NdrFcShort( 0x164 ), /* Offset=
356 (794) */
/* 440 */ NdrFcLong( 0x14 ), /* 20 */
/* 444 */ NdrFcShort( 0x17a ), /* Offset=
378 (822) */
/* 446 */ NdrFcShort( 0xffff ), /* Offset= -1
(445) */
/* 448 */
FC_BOGUS_ARRAY */
0x21, /*
0x3, /*
3 */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 454 */ NdrFcShort( 0x0 ), /* 0 */
/* 456 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 458 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 462 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 464 */
0x12, 0x0, /*
FC_UP */
/* 466 */ NdrFcShort( 0xff74 ), /* Offset= -
140 (326) */
/* 468 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */

```

```

/* 470 */
FC_BOGUS_STRUCT */
0x1a, /*
0x3, /*
3 */
/* 472 */ NdrFcShort( 0x10 ), /* 16 */
/* 474 */ NdrFcShort( 0x0 ), /* 0 */
/* 476 */ NdrFcShort( 0x6 ), /* Offset= 6 (482) */
/* 478 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 480 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 482 */
0x11, 0x0, /*
FC_RP */
/* 484 */ NdrFcShort( 0xffdc ), /* Offset= -
36 (448) */
/* 486 */
FC_BOGUS_ARRAY */
0x21, /*
0x3, /*
3 */
/* 488 */ NdrFcShort( 0x0 ), /* 0 */
/* 490 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 492 */ NdrFcShort( 0x0 ), /* 0 */
/* 494 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 496 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 500 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 502 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
*/
/* 504 */ NdrFcShort( 0xff58 ), /* Offset= -
168 (336) */
/* 506 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 508 */
0x1a, /*
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( 0xffdc ), /* Offset= -
36 (486) */
/* 524 */

```

```

FC_BOGUS_ARRAY */
0x21, /*
0x3, /*
3 */
/* 526 */ NdrFcShort( 0x0 ), /* 0 */
/* 528 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 534 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 538 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 540 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 542 */ NdrFcShort( 0xff44 ), /* Offset= -
188 (354) */
/* 544 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 546 */
FC_BOGUS_STRUCT */
0x1a, /*
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( 0xffdc ), /* Offset= -
36 (524) */
/* 562 */
FC_BOGUS_ARRAY */
0x21, /*
0x3, /*
3 */
/* 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( 0xffdc ), /* Offset= -
36 (562) */
/* 600 */
0x2f, /*
FC_IP */
0x5a, /*
FC_CONSTANT_IID */
/* 602 */ NdrFcLong( 0x2f ), /* 47 */
/* 606 */ NdrFcShort( 0x0 ), /* 0 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ 0xc0, /* 192 */
0x0, /*
0 */
/* 612 */ 0x0, /* 0 */
0x0, /*
0 */
/* 614 */ 0x0, /* 0 */
0x0, /*
0 */
/* 616 */ 0x0, /* 0 */
0x46, /*
70 */
/* 618 */
0x1b, /*
FC_CARRAY */
0x0, /*
0 */
/* 620 */ NdrFcShort( 0x1 ), /* 1 */
/* 622 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 624 */ NdrFcShort( 0x4 ), /* 4 */
/* 626 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 628 */ 0x1, /* FC_BYTE */
0x5b, /*
FC_END */
/* 630 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 632 */ NdrFcShort( 0x18 ), /* 24 */

```

```

/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0xa ), /* Offset= 10 (646) */
/* 638 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 640 */ 0x4c, /* FC_EMBEDDED_COMPLEX
*/
0x0, /*
0 */
/* 642 */ NdrFcShort( 0xffd6 ), /* Offset= -
42 (600) */
/* 644 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 646 */
0x12, 0x0, /*
FC_UP */
/* 648 */ NdrFcShort( 0xffe2 ), /* Offset= -
30 (618) */
/* 650 */
0x21, /*
FC_BOGUS_ARRAY */
0x3, /*
3 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 656 */ NdrFcShort( 0x0 ), /* 0 */
/* 658 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 660 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 664 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 666 */
0x12, 0x0, /*
FC_UP */
/* 668 */ NdrFcShort( 0xffda ), /* Offset= -
38 (630) */
/* 670 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 672 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /*
3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ NdrFcShort( 0x6 ), /* Offset= 6 (684) */
/* 680 */ 0x8, /* FC_LONG */
0x40, /*
FC_STRUCTPAD4 */
/* 682 */ 0x36, /* FC_POINTER */
0x5b, /*
FC_END */
/* 684 */
0x11, 0x0, /*
FC_RP */
/* 686 */ NdrFcShort( 0xffdc ), /* 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( 0xffff1 ),
/* 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( 0xffe7 ),
/* Offset= -25 (694) */
0x5b, /*
FC_END */
/* 722 */
0x11, 0x0, /*
FC_RP */
/* 724 */ NdrFcShort( 0xff12 ), /* Offset= -
238 (486) */
/* 726 */
0x1b, /*
FC_CARRAY */
0x0, /*
0 */
/* 728 */ NdrFcShort( 0x1 ), /* 1 */
/* 730 */ 0x19, /* Corr desc: field
pointer, FC_ULONG */
0x0, /*
*/
/* 732 */ NdrFcShort( 0x0 ), /* 0 */
/* 734 */ NdrFcShort( 0x1 ), /* Corr flags: early,
*/
/* 736 */ 0x1, /* FC_BYTE */

```

FC_END */ /*_738 */	0x5b,	/*	/* 786 */ 0x19, pointer, FC_ULONG */	/* Corr desc: field 0x0,	/*	/* 836 */ NdrFcShort(0xffe6),	/* Offset= - 26 (810) */
FC_BOGUS_STRUCT */	0x1a,	/*	/*	/* 788 */ NdrFcShort(0x0), /* 0 */	/*	FC_STRUCT */	0x15,
3 */	0x3,	/*	/* 790 */ NdrFcShort(0x1), /* Corr flags: early, */	/* 792 */ 0x8,	/* FC_LONG */	0x3,	/*
/* 740 */ NdrFcShort(0x10), /* 16 */			/* 794 */	FC_END */	/*	3 */	/* 840 */ NdrFcShort(0x8), /* 8 */
/* 742 */ NdrFcShort(0x0), /* 0 */			FC_BOGUS_STRUCT */	/*_794 */	/*	/* 842 */ 0x8,	/* FC_LONG */
/* 744 */ NdrFcShort(0x6), /* Offset= 6 (750) */			3 */	0x1a,	/*	FC_LONG */	0x8,
/* 746 */ 0x8,	0x40,	/*	/* 796 */ NdrFcShort(0x10), /* 16 */	FC_BOGUS_STRUCT */	/*	/*_844 */ 0x5c,	/* FC_PAD */
FC_STRUCTPAD4 */	/* FC_POINTER */	/*	/* 798 */ NdrFcShort(0x0), /* 0 */	0x3,	/*	FC_END */	0x5b,
/*_748 */ 0x36,	0x5b,	/*	/* 800 */ NdrFcShort(0x6), /* Offset= 6 (806) */	3 */	/*	/*_846 */	/*
FC_END */	0x12, 0x0,	/*	/* 802 */ 0x8,	/* 796 */ NdrFcShort(0x10), /* 16 */	/*	FC_CARRAY */	0x1b,
/*_750 */			FC_STRUCTPAD4 */	/* 798 */ NdrFcShort(0x0), /* 0 */	/*	0x3,	/*
FC_UP */	0x1b,	/*	/*_804 */ 0x36,	/* 800 */ NdrFcShort(0x6), /* Offset= 6 (806) */	/*	3 */	/* 848 */ NdrFcShort(0x8), /* 8 */
/*_752 */ NdrFcShort(0xffe6), /* Offset= - 26 (726) */			FC_END */	/* 802 */ 0x8,	/* FC_LONG */	/*	/*_850 */ 0x7,
/*_754 */	0x1b,	/*	/*_806 */	/*_804 */ 0x36,	/* FC_POINTER */	/*	/* Corr desc: FC_USHORT */
FC_CARRAY */	0x1,	/*	FC_UP */	FC_END */	/*	0x0,	/*
1 */			/*_808 */	/*_806 */	/*	/*_852 */ NdrFcShort(0xffc8),	/* -56 */
/* 756 */ NdrFcShort(0x2), /* 2 */			FC_CARRAY */	0x12, 0x0,	/*	/*_854 */ NdrFcShort(0x1), /* Corr flags: early, */	/*
/* 758 */ 0x19,			7 */	FC_UP */	/*	/*_856 */ 0x4c,	/* FC_EMBEDDED_COMPLEX */
/* Corr desc: field pointer, FC_ULONG */	0x0,	/*	/* 812 */ NdrFcShort(0x8), /* 8 */	/*_808 */ NdrFcShort(0xffe6), /* Offset= - 26 (782) */	/*	0x0,	/*
/*			/* 814 */ 0x19,	/*_810 */	/*	0x1b,	/*
/* 760 */ NdrFcShort(0x0), /* 0 */			/* Corr desc: field pointer, FC_ULONG */	FC_CARRAY */	/*	0x7,	/*
/* 762 */ NdrFcShort(0x1), /* Corr flags: early, */			/*	7 */	/*	/*_816 */ NdrFcShort(0x0), /* 0 */	/*
/* 764 */ 0x6,	0x5b,	/*	/* 816 */ NdrFcShort(0x0), /* 0 */	/*_812 */ NdrFcShort(0x8), /* 8 */	/*	/*_818 */ NdrFcShort(0x1), /* Corr flags: early, */	/*
FC_END */	0x1a,	/*	/*_820 */ 0xb,	/*_814 */ 0x19,	/* FC_HYPER */	/*	/*_820 */ NdrFcShort(0x0), /* 0 */
/*_766 */	0x3,	/*	FC_END */	/* Corr desc: field pointer, FC_ULONG */	/* FC_LONG */	/*	/*_818 */ NdrFcShort(0x1), /* Corr flags: early, */
FC_BOGUS_STRUCT */			/*_822 */	0x0,	/*	0x5b,	/*
3 */			/*	/*	/*	0x1a,	/*
/* 768 */ NdrFcShort(0x10), /* 16 */			/*_816 */ NdrFcShort(0x0), /* 0 */	/*_822 */	/*	FC_BOGUS_STRUCT */	0x3,
/* 770 */ NdrFcShort(0x0), /* 0 */			/*_818 */ NdrFcShort(0x1), /* Corr flags: early, */	FC_CARRAY */	/*	3 */	/*
/* 772 */ NdrFcShort(0x6), /* Offset= 6 (778) */			/*_820 */ 0xb,	7 */	/*	/*_864 */ NdrFcShort(0x38), /* 56 */	/*
/* 774 */ 0x8,	0x40,	/*	FC_END */	/*_812 */ NdrFcShort(0x8), /* 8 */	/*	/*_866 */ NdrFcShort(0xffec), /* Offset= - 20 (846) */	/*
/* FC_LONG */	FC_STRUCTPAD4 */	/*	/*_824 */	/*_814 */ 0x19,	/* Corr desc: field pointer, FC_ULONG */	/*_868 */ NdrFcShort(0x0), /* Offset= 0 (868) */	/*
0x40,	/* FC_POINTER */	/*	/*_826 */	/* Corr desc: field pointer, FC_ULONG */	/*	/*_870 */ 0x6,	/* FC_SHORT */
FC_STRUCTPAD4 */	0x5b,	/*	FC_BOGUS_STRUCT */	0x1a,	/*	FC_SHORT */	0x6,
/*_776 */ 0x36,	0x5b,	/*	3 */	0x3,	/*	/*_872 */ 0x8,	/* FC_LONG */
FC_END */			/* 824 */ NdrFcShort(0x10), /* 16 */	0x12, 0x0,	/*	0x8,	/*
/*_778 */	0x12, 0x0,	/*	/* 826 */ NdrFcShort(0x0), /* 0 */	FC_END */	/*	FC_LONG */	/*
FC_UP */			/*_828 */ NdrFcShort(0x6), /* Offset= 6 (834) */	/*_778 */	/*	/*_874 */ 0x40,	/* FC_STRUCTPAD4 */
/*_780 */ NdrFcShort(0xffe6), /* Offset= - 26 (754) */			/*_830 */ 0x8,	/*_780 */ NdrFcShort(0xffe6), /* Offset= - 26 (754) */	/*	0x4c,	/*
/*_782 */	0x1b,	/*	FC_END */	/*_782 */	/*	FC_EMBEDDED_COMPLEX */	/*
FC_CARRAY */	0x3,	/*	/*_832 */ 0x36,	/*_784 */	/*	/*_876 */ 0x0,	/* 0 */
3 */			/* FC_POINTER */	/*_786 */	/*	NdrFcShort(0xfe0f),	/*
/* 784 */ NdrFcShort(0x4), /* 4 */			FC_END */	/*_788 */	/*	/* Offset= -497 (380) */	/*
			/*_834 */	FC_CARRAY */	/*	0x5b,	/*
			0x12, 0x0,	3 */	/*	FC_END */	/*
			/*	/*_784 */ NdrFcShort(0x4), /* 4 */	/*	/*_880 */	/*
			FC_UP */		/*	0x12, 0x0,	/*

```

/* 882 */ NdrFcShort( 0xff04 ), /* Offset= -
252 (630) */
/* 884 */
FC_UP [simple_pointer] */
/* 886 */ 0x1, /* FC_BYTE */
FC_PAD */
/* 888 */
FC_UP [simple_pointer] */
/* 890 */ 0x6, /* FC_SHORT */
FC_PAD */
/* 892 */
FC_UP [simple_pointer] */
/* 894 */ 0x8, /* FC_LONG */
FC_PAD */
/* 896 */
FC_UP [simple_pointer] */
/* 898 */ 0xb, /* FC_HYPER */
FC_PAD */
/* 900 */
FC_UP [simple_pointer] */
/* 902 */ 0xa, /* FC_FLOAT */
FC_PAD */
/* 904 */
FC_UP [simple_pointer] */
/* 906 */ 0xc, /* FC_DOUBLE */
FC_PAD */
/* 908 */
FC_UP */
/* 910 */ NdrFcShort( 0xfda2 ), /* Offset= -
606 (304) */
/* 912 */
FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xfda4 ), /* Offset= -
604 (310) */
/* 916 */
FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfdba ), /* Offset= -
582 (336) */
/* 920 */
FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0xfdc8 ), /* Offset= -
568 (354) */
/* 924 */
FC_UP [pointer_deref] */
/* 926 */ NdrFcShort( 0xfdd6 ), /* Offset= -
554 (372) */

```

```

/* 928 */
FC_UP [pointer_deref] */
/* 930 */ NdrFcShort( 0x2 ), /* Offset= 2 (932) */
/* 932 */
FC_UP */
/* 934 */ NdrFcShort( 0x14 ), /* Offset= 20 (954) */
/* 936 */
FC_STRUCT */
7 */
/* 938 */ NdrFcShort( 0x10 ), /* 16 */
/* 940 */ 0x6, /* FC_SHORT */
FC_BYTE */
/* 942 */ 0x1, /* FC_BYTE */
FC_LONG */
/* 944 */ 0xb, /* FC_HYPER */
FC_END */
/* 946 */
FC_UP */
/* 948 */ NdrFcShort( 0xffff4 ), /* Offset= -
12 (936) */
/* 950 */
FC_UP [simple_pointer] */
/* 952 */ 0x2, /* FC_CHAR */
FC_PAD */
/* 954 */
FC_BOGUS_STRUCT */
7 */
/* 956 */ NdrFcShort( 0x20 ), /* 32 */
/* 958 */ NdrFcShort( 0x0 ), /* 0 */
/* 960 */ NdrFcShort( 0x0 ), /* Offset= 0 (960) */
/* 962 */ 0x8, /* FC_LONG */
FC_LONG */
/* 964 */ 0x6, /* FC_SHORT */
FC_SHORT */
/* 966 */ 0x6, /* FC_SHORT */
FC_SHORT */
/* 968 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0 */
/* 970 */ NdrFcShort( 0xfc3c ), /* Offset= -
964 (6) */
/* 972 */ 0x5c, /* FC_PAD */
FC_END */
/* 974 */ 0xb4, /* FC_USER_MARSHAL */
131 */

```

```

/* 976 */ NdrFcShort( 0x0 ), /* 0 */
/* 978 */ NdrFcShort( 0x18 ), /* 24 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0xfc2c ), /* Offset= -
980 (2) */
/* 984 */
FC_RP [allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /* Offset= 6 (992) */
/* 988 */
FC_OP */
/* 990 */ NdrFcShort( 0xffdc ), /* Offset= -
36 (954) */
/* 992 */ 0xb4, /* FC_USER_MARSHAL */
131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xffff4 ), /*
Offset= -12 (988) */
};
static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};
/* Standard interface: __MIDL_itf_tpc_com_ps_0000,
ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00}} */
/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0
x00,0x00,0x00,0x46}} */
/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0
x4F,0xBF,0xE0,0x8B}} */
#pragma code_seg(".orpc")
static const unsigned short
ITPCC_FormatStringOffsetTable[] =

```



```

// 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
int
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, LPCSTR, LPCSTR, LPCSTR,
DBUSMALLINT);

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int
msgstate, int severity,
LPCSTR
msgtext, LPCSTR srvername, 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 pDest 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, "" }
    };

    return errorMsgs[i];
}

```

```

        static char szNotFound[] = "Unknown error
number.";
        for(i=0; errorMsgs[i].szMsg[0]; i++)
        {
            if ( m_errno ==
errorMsgs[i].iError )
                break;
        }
        if ( !errorMsgs[i].szMsg[0] )
            return szNotFound;
        else
            return errorMsgs[i].szMsg;
    }

    // wrapper routine for class constructor
    __declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
        LPCSTR szServer,          // name of
SQL server
        LPCSTR szUser,           //
user name for login
        LPCSTR szPassword,       // password
for login
        LPCSTR szHost,          //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
        LPCSTR szDatabase )      // name of
database to use
    {
        return new CTPCC_DBLIB( szServer, szUser,
szPassword, szHost, szDatabase );
    }

    CTPCC_DBLIB::CTPCC_DBLIB (
        LPCSTR szServer,        // name of
SQL server
        LPCSTR szUser,          //
user name for login
        LPCSTR szPassword,      // password
for login
        LPCSTR szHost,          //
workstation name; shows up in sp_who; max 30 chars,
only first 10 kept by SQL Server
        LPCSTR szDatabase )     // name of
database to use
    {
        LOGINREC *login;
        const BYTE *pData;

        // initialization
        m_dbproc = NULL;
        m_DbLibErr = (CDBLIBERR*)NULL;
        m_SqlErr = (CSQLERR*)NULL;

        m_MaxRetries = 10;      // how many
retries on deadlock

        // increase max number of connections if
getting close
        if ( dbgetmaxprocs() < (iConnectionCount+5)
)
            {

```

```

                if (
dbsetmaxprocs(iConnectionCount+10) == FAIL )
                    ThrowError(CDBLIBERR::eDbSetMaxProcs);
            }

            // allocate a login structure
            login = dblogin();
            if (login == NULL)
                ThrowError(CDBLIBERR::eLogin);
            InterlockedIncrement( &iConnectionCount );

            // register error and message handler
        functions
            if (dbprocerrhandle(login, err_handler) ==
NULL)
                ThrowError(CDBLIBERR::eDbProcHandler);

            if (dbprocmsghandle(login, msg_handler) ==
NULL)
                ThrowError(CDBLIBERR::eDbProcHandler);

            DBSETLUSER(login, szUser);
            DBSETLPWD(login, szPassword);
            DBSETHOST(login, szHost);
            DBSETLPACKET(login, (unsigned
short)DEFCLPACKSIZE);
            DBSETLVERSION(login, DBVER60);
            // use dblib ver 6.0 client behavior

            // set time to wait for login
            if (dbsetlogintime(60) == FAIL)
                ThrowError(CDBLIBERR::eDbSet);

            // set time to wait for statement execution
            if (dbsettime(180) == FAIL)
                ThrowError(CDBLIBERR::eDbSet);

            m_dbproc = dbopen(login, szServer);

            // deallocate login structure before
checking for success
            dbfreelogin( login );

            if (m_dbproc == NULL)
                ThrowError(CDBLIBERR::eDbOpen);

            // save address of class instance so that
the message and error handler
            // can get to data.
            dbsetuserdata(m_dbproc, (LPVOID)this);

            // Use the the right database
            if (dbuse(m_dbproc, szDatabase) == FAIL)
                ThrowError(CDBLIBERR::eDbUse);

            dbcmd(m_dbproc, "set nocount on ");
            // do not return row counts
            dbcmd(m_dbproc, "set XACT_ABORT ON");
            // rollback transaction on abort

```

```

            if (dbsqlxexec(m_dbproc) == FAIL)
                ThrowError(CDBLIBERR::eDbSqlExec);
            DiscardNextResults(2);

            // verify that version of stored procs on
server is correct
            dbrpcinit(m_dbproc, "tpcc_version", 0);
            if (dbrpcexec(m_dbproc) == FAIL)
                ThrowError(CDBLIBERR::eDbRpcExec);

            if (dbresults(m_dbproc) != SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

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

            char szSrvVersion[16];
            pData=dbdata(m_dbproc, 1);
            if (pData)
                UtilStrCpy(szSrvVersion, pData,
dbdatalen(m_dbproc, 1));
            else
                szSrvVersion[0]=0;
            if (strcmp(szSrvVersion,sVersion))
                throw new CTPCC_DBLIB_ERR(
CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );

            DiscardNextRows(0);
            DiscardNextResults(0);
        }

        CTPCC_DBLIB::~CTPCC_DBLIB( void )
        {
            // close db connection and deallocate
resources
            dbclose(m_dbproc);
            InterlockedDecrement( &iConnectionCount );
            if (m_DbLibErr != NULL)
                delete m_DbLibErr;
            if (m_SqlErr != NULL)
                delete m_SqlErr;
        }

        void CTPCC_DBLIB::SetDbLibError(int severity, int
dberr, int oserr, LPCSTR dberstr, LPCSTR oserrstr)
        {
            delete m_DbLibErr;
            m_DbLibErr = new
CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr,
oserr);

            if (dberstr != 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::eDbNextRow);
            else
                break;
        }
        iRowsRead++;
    }

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

// Read and discard results until no more. Throw an
exception if number of result sets read doesn't
// match number expected. The result set count will
be ignored if the expected count value
// passed in is negative. A typical use of this
routine is to verify that there are no more
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int
iExpectedCount)
{
    int iResultsRead = 0;
    RETCODE rc;

    while (TRUE)
    {
        rc = dbresults(m_dbproc);
        if (rc == NO_MORE_RESULTS)

```

```

            break;
        if (rc == FAIL)
        {
            if (iExpectedCount >=
0)
                ThrowError(CDBLIBERR::eDbResults);
            else
                break;
        }

        DiscardNextRows(-1);
        iResultsRead++;
    }

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

void CTPCC_DBLIB::StockLevel()
{
    int iTryCount =
0;
    const BYTE *pData;
    ResetError();

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

            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.StockLevel.w_id); // @w_id int
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.StockLevel.d_id); // @d_id
            tinyint
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.StockLevel.threshold); // @threshold
            smallint

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

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

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

```



```

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

        DiscardNextRows(0);
        DiscardNextResults(0);

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

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

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

    int                iTryCount =
0;
    const BYTE         *pData;

    ResetError();

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

```

```

            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.w_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.d_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.c_id);
            dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.NewOrder.o_ol_cnt);

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

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

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

            for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; i++)
            {
                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_i_id);
                dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_supply_w_id);
                dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1,
-1, (BYTE *) &m_txn.NewOrder.OL[i].ol_quantity);
            }

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

            // Get order line
            results

            m_txn.NewOrder.total_amount = 0;
            for (i = 0;
i<m_txn.NewOrder.o_ol_cnt; i++)
            {

```

```

            if
(dbresults(m_dbproc) != SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);

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

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

            if (pData=dbdata(m_dbproc, 1))
                UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,
pData, dbdatlen(m_dbproc, 1));

            if (pData=dbdata(m_dbproc, 2))
                m_txn.NewOrder.OL[i].ol_stock =
                (*DBSMALLINT *) pData);
            if (pData=dbdata(m_dbproc, 3))
                UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_ge
neric, 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)
!= SUCCEEDED)
            ThrowError(CDBLIBERR::eDbResults);

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

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

        if
(pData=dbdata(m_dbproc, 1))

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

        if
(pData=dbdata(m_dbproc, 2))

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

        if
(pData=dbdata(m_dbproc, 3))

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

        if
(pData=dbdata(m_dbproc, 4))

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

        if
(pData=dbdata(m_dbproc, 5))

            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8, (BYTE
*)&m_txn.NewOrder.c_discount, 8);

        if
(pData=dbdata(m_dbproc, 6))

            UtilStrCpy(m_txn.NewOrder.c_credit, pData,
dbdatlen(m_dbproc, 6));

        if
(pData=dbdata(m_dbproc, 7))

            {
                datetime =
                (*(DBDATETIME *) pData);

                dbdatecrack(m_dbproc, &daterec, &datetime);

                m_txn.NewOrder.o_entry_d.year =
                daterec.year;

```

```

            m_txn.NewOrder.o_entry_d.month =
            daterec.month;

            m_txn.NewOrder.o_entry_d.day =
            daterec.day;

            m_txn.NewOrder.o_entry_d.hour =
            daterec.hour;

            m_txn.NewOrder.o_entry_d.minute =
            daterec.minute;

            m_txn.NewOrder.o_entry_d.second =
            daterec.second;

            if
            (pData=dbdata(m_dbproc, 8))

                commit_flag =
                (*(DBTINYINT *) pData);

                DiscardNextRows(0);
                DiscardNextResults(0);

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

                    m_txn.NewOrder.exec_status_code = eOK;
                }
                else
                {
                    m_txn.NewOrder.exec_status_code =
                    eInvalidItem;

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

                }
                else
                {
                    throw;
                }
            }
            // while (TRUE)

```

```

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

void CTPCC_DBLIB::Payment()
{
    DBDATETIME          datetime;
    DBDATERECD          daterec;

    int                  iTryCount =
    0;

    const BYTE          *pData;

    ResetError();

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

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

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

            dbrpcparam(m_dbproc,
            NULL, 0, SQLFLT8, -1, -1, (BYTE *)
            &m_txn.Payment.h_amount);

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

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

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

            // if customer id is
            zero, then payment is by name
            if (m_txn.Payment.c_id
            == 0)

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

            if (dbrpcexec(m_dbproc)
            == FAIL)

                ThrowError(CDBLIBERR::eDbRpcExec);

            if (dbresults(m_dbproc)
            != SUCCEEDED)

                ThrowError(CDBLIBERR::eDbResults);

```

```

        if (dbnextrow(m_dbproc)
!= REG_ROW)
            ThrowError(CDBLIBERR::eDbNextRow);
        if (dbnumcols(m_dbproc)
!= 27)
            ThrowError(CDBLIBERR::eWrongNumCols);
        if
(pData=dbdata(m_dbproc, 1))
            m_txn.Payment.c_id = *((DBINT *) pData);
        if
(pData=dbdata(m_dbproc, 2))
            UtilStrCpy(m_txn.Payment.c_last, pData,
dbdatlen(m_dbproc, 2));
        if
(pData=dbdata(m_dbproc, 3))
        {
            datetime =
*(DBDATETIME *) pData);
            dbdatecrack(m_dbproc, &daterec, &datetime);
            m_txn.Payment.h_date.year = daterec.year;
            m_txn.Payment.h_date.month =
daterec.month;
            m_txn.Payment.h_date.day = daterec.day;
            m_txn.Payment.h_date.hour = daterec.hour;
            m_txn.Payment.h_date.minute =
daterec.minute;
            m_txn.Payment.h_date.second =
daterec.second;
        }
        if
(pData=dbdata(m_dbproc, 4))
            UtilStrCpy(m_txn.Payment.w_street_1, pData,
dbdatlen(m_dbproc, 4));
        if
(pData=dbdata(m_dbproc, 5))
            UtilStrCpy(m_txn.Payment.w_street_2, pData,
dbdatlen(m_dbproc, 5));
        if
(pData=dbdata(m_dbproc, 6))
            UtilStrCpy(m_txn.Payment.w_city, pData,
dbdatlen(m_dbproc, 6));
        if
(pData=dbdata(m_dbproc, 7))
            UtilStrCpy(m_txn.Payment.w_state, pData,
dbdatlen(m_dbproc, 7));

```

```

        if
(pData=dbdata(m_dbproc, 8))
            UtilStrCpy(m_txn.Payment.w_zip, pData,
dbdatlen(m_dbproc, 8));
        if
(pData=dbdata(m_dbproc, 9))
            UtilStrCpy(m_txn.Payment.d_street_1, pData,
dbdatlen(m_dbproc, 9));
        if
(pData=dbdata(m_dbproc, 10))
            UtilStrCpy(m_txn.Payment.d_street_2, pData,
dbdatlen(m_dbproc, 10));
        if
(pData=dbdata(m_dbproc, 11))
            UtilStrCpy(m_txn.Payment.d_city, pData,
dbdatlen(m_dbproc, 11));
        if
(pData=dbdata(m_dbproc, 12))
            UtilStrCpy(m_txn.Payment.d_state, pData,
dbdatlen(m_dbproc, 12));
        if
(pData=dbdata(m_dbproc, 13))
            UtilStrCpy(m_txn.Payment.d_zip, pData,
dbdatlen(m_dbproc, 13));
        if
(pData=dbdata(m_dbproc, 14))
            UtilStrCpy(m_txn.Payment.c_first, pData,
dbdatlen(m_dbproc, 14));
        if
(pData=dbdata(m_dbproc, 15))
            UtilStrCpy(m_txn.Payment.c_middle, pData,
dbdatlen(m_dbproc, 15));
        if
(pData=dbdata(m_dbproc, 16))
            UtilStrCpy(m_txn.Payment.c_street_1, pData,
dbdatlen(m_dbproc, 16));
        if
(pData=dbdata(m_dbproc, 17))
            UtilStrCpy(m_txn.Payment.c_street_2, pData,
dbdatlen(m_dbproc, 17));
        if
(pData=dbdata(m_dbproc, 18))
            UtilStrCpy(m_txn.Payment.c_city, pData,
dbdatlen(m_dbproc, 18));
        if
(pData=dbdata(m_dbproc, 19))
            UtilStrCpy(m_txn.Payment.c_state, pData,
dbdatlen(m_dbproc, 19));
        if
(pData=dbdata(m_dbproc, 20))

```

```

            UtilStrCpy(m_txn.Payment.c_zip, pData,
dbdatlen(m_dbproc, 20));
        if
(pData=dbdata(m_dbproc, 21))
            UtilStrCpy(m_txn.Payment.c_phone, pData,
dbdatlen(m_dbproc, 21));
        if
(pData=dbdata(m_dbproc, 22))
        {
            datetime =
*(DBDATETIME *) pData);
            dbdatecrack(m_dbproc, &daterec, &datetime);
            m_txn.Payment.c_since.year =
daterec.year;
            m_txn.Payment.c_since.month =
daterec.month;
            m_txn.Payment.c_since.day = daterec.day;
            m_txn.Payment.c_since.hour =
daterec.hour;
            m_txn.Payment.c_since.minute =
daterec.minute;
            m_txn.Payment.c_since.second =
daterec.second;
        }
        if (pData=dbdata(m_dbproc, 23))
            UtilStrCpy(m_txn.Payment.c_credit, pData,
dbdatlen(m_dbproc, 23));
        if (pData=dbdata(m_dbproc, 24))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8, (BYTE
*)&m_txn.Payment.c_credit_lim, 8);
        if (pData=dbdata(m_dbproc, 25))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8, (BYTE
*)&m_txn.Payment.c_discount, 8);
        if (pData=dbdata(m_dbproc, 26))
            dbconvert(m_dbproc, SQLNUMERIC,
(LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8, (BYTE
*)&m_txn.Payment.c_balance, 8);
        if (pData=dbdata(m_dbproc, 27))
            UtilStrCpy(m_txn.Payment.c_data, pData,
dbdatlen(m_dbproc, 27));
        DiscardNextRows(0);

```

```

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

void CTPCC_DBLIB::OrderStatus()
{
    int i;
    DBDATETIME datetime;
    DBDATERECD daterec;

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

    ResetError();

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

```

```

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

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

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

            // Get order lines
            if (dbresults(m_dbproc)
!= SUCCEED)
            {
                if
                ((m_DbLibErr == NULL) && (m_SqlErr == NULL))
                    throw new CTPCC_DBLIB_ERR(
                    CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                else
                    ThrowError(CDBLIBERR::eDbResults);
            }

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

            i = 0;
            while (TRUE)
            {
                rc =
                dbnextrow(m_dbproc);
                if (rc ==
                NO_MORE_ROWS)
                    break;
                if (rc !=
                REG_ROW)
                    ThrowError(CDBLIBERR::eDbNextRow);

                if (pData=dbdata(m_dbproc, 1))
                    m_txn.OrderStatus.OL[i].ol_supply_w_id =
                    (*(DBSMALLINT *) pData);

```

```

                if (pData=dbdata(m_dbproc, 2))
                    m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT
                    *) pData);
                if (pData=dbdata(m_dbproc, 3))
                    m_txn.OrderStatus.OL[i].ol_quantity =
                    (*(DBSMALLINT *) pData);
                if (pData=dbdata(m_dbproc, 4))
                    dbconvert(m_dbproc, SQLNUMERIC,
                    (LPCBYTE)pData, dbdatlen(m_dbproc,4),
                    SQLFLT8, (BYTE
                    *)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                if (pData=dbdata(m_dbproc, 5))
                {
                    datetime = (*(DBDATETIME *) pData);
                    dbdatecrack(m_dbproc, &daterec, &datetime);
                    m_txn.OrderStatus.OL[i].ol_delivery_d.year
                    = daterec.year;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.month
                    = daterec.month;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.day
                    = daterec.day;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.hour
                    = daterec.hour;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.minut
                    e = daterec.minute;
                    m_txn.OrderStatus.OL[i].ol_delivery_d.secon
                    d = daterec.second;
                }
                i++;
            }
            m_txn.OrderStatus.o_ol_cnt = i;

            if (dbresults(m_dbproc)
!= SUCCEED)
                ThrowError(CDBLIBERR::eDbResults);
            if (dbnextrow(m_dbproc)
!= REG_ROW)
                ThrowError(CDBLIBERR::eDbNextRow);
            if (dbnumcols(m_dbproc)
!= 8)

```

```

ThrowError(CDBLIBERR::eWrongNumCols);

if (pData=dbdata(m_dbproc, 1))
    m_txn.OrderStatus.c_id = *(DBINT *)
pData);

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

if (pData=dbdata(m_dbproc, 3))
    UtilStrCpy(m_txn.OrderStatus.c_first,
pData, dbdatlen(m_dbproc,3));

if (pData=dbdata(m_dbproc, 4))
    UtilStrCpy(m_txn.OrderStatus.c_middle,
pData, dbdatlen(m_dbproc, 4));

if (pData=dbdata(m_dbproc, 5))
    {
        datetime =
*(DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.OrderStatus.o_entry_d.year =
daterec.year;
        m_txn.OrderStatus.o_entry_d.month =
daterec.month;
        m_txn.OrderStatus.o_entry_d.day =
daterec.day;
        m_txn.OrderStatus.o_entry_d.hour =
daterec.hour;
        m_txn.OrderStatus.o_entry_d.minute =
daterec.minute;
        m_txn.OrderStatus.o_entry_d.second =
daterec.second;
    }

if (pData=dbdata(m_dbproc, 6))
    m_txn.OrderStatus.o_carrier_id =
*(DBSMALLINT *) pData);

if (pData=dbdata(m_dbproc, 7))
    dbconvert(m_dbproc, SQLNUMERIC,
(LPBYTE)pData, dbdatlen(m_dbproc,7),
SQLFLT8, (BYTE
*)&m_txn.OrderStatus.c_balance, 8);

```

```

if (pData=dbdata(m_dbproc, 8))
    m_txn.OrderStatus.o_id = *(DBINT *)
pData);

DiscardNextRows(0);
DiscardNextResults(0);

if
(m_txn.OrderStatus.o_ol_cnt == 0)
    throw new
CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER
);

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

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

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

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

    ResetError();

```

```

while (TRUE)
{
    try
    {
        dbrpcinit(m_dbproc,
"tpcc_delivery", 0);
        dbrpcparam(m_dbproc,
NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.Delivery.w_id);
        dbrpcparam(m_dbproc,
NULL, 0, SQLINT1, -1, -1, (BYTE *)
&m_txn.Delivery.o_carrier_id);

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

        if (dbresults(m_dbproc)
!= SUCCEED)
            ThrowError(CDBLIBERR::eDbResults);

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

        if (dbnumcols(m_dbproc)
!= 10)
            ThrowError(CDBLIBERR::eWrongNumCols);

        for (i=0; i<10; i++)
        {
            if (pData =
dbdata(m_dbproc, i+1))
                m_txn.Delivery.o_id[i] = *(DBINT *)pData);
        }

        DiscardNextRows(0);
        DiscardNextResults(0);

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

```

```

        delete e;
        Sleep(10 *
iTryCount);
    }
    else
        throw;
    }
    // while (TRUE)
//
// if (iTryCount)
//     throw new
CTPCC_DBLIB_ERR(CTPCC_DBLIB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_DBLIB::ResetError()
{
    if (m_DbLibErr != NULL)
    {
        delete m_DbLibErr;
        m_DbLibErr = (CDBLIBERR*)NULL;
    }

    if (m_SqlErr != NULL)
    {
        delete m_SqlErr;
        m_SqlErr = (CSQLERR*)NULL;
    }

    return;
}

```

tpcc_odbc.cpp

```

/*
 * FILE: TPC_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 COMPILER_FOR_SNAC // define that to
// compile for SQL Native Client; comment out to use
// MDAC

#ifdef COMPILER_FOR_SNAC
#include <odbcss.h>
#else
// Compile for SNAC
#include <sqlncli.h>
#endif

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

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

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

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

const iMaxRetries = 3; // how many
retries on deadlock
//const iMaxRetries = 0; // for
// debugging

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

static SQLHENV henv = SQL_NULL_HENV;
// ODBC environment handle

BOOL WINAPI 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." },
        { ERR_INVALID_NEW_ORDER_PARAM,
"New Order parameter invalid." },
        { 0, "" }
    };

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

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno ==
errorMsgs[i].iError )
            break;
        if ( !errorMsgs[i].szMsg[0] )
            return szNotFound;
        else
            return errorMsgs[i].szMsg;
    }

    // wrapper routine for class constructor
    __declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
LPCSTR szServer, // name of
SQL server
LPCSTR szUser, //
user name for login
LPCSTR szPassword, // password
for login

```

```

        LPCSTR szHost,                //
not used
        LPCSTR szDatabase,           // name of
database to use
        LPCWSTR szSPPrefix,         // prefix to
append to the stored procedure names
        BOOL bCallNoDuplicatesNewOrder ) // whether
to check for non-duplicate items in NewOrder and call
a new SP
{
    return new CTPCC_ODBC( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix,
bCallNoDuplicatesNewOrder );
}

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,
    // name of SQL server
    LPCSTR szUser,
    // user name for login
    LPCSTR szPassword,
    // password for login
    LPCSTR szHost,
    // not used
    LPCSTR szDatabase,
    // name of database to use
    LPCWSTR szSPPrefix,
    // prefix to append to the stored procedure
names
    BOOL bCallNoDuplicatesNewOrder //
whether to check for non-duplicate items in NewOrder
and call a new SP
)
:
m_bCallNoDuplicatesNewOrder(bCallNoDuplicatesNewOrder)
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

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

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

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

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

        ThrowError(CODBCERR::eAllocHandle);
}

```

```

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

            ThrowError(CODBCERR::eConnOption);

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

#ifdef COMPILE_FOR_SNAC
            sprintf( szConnectStr,
"DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
szServer, szUser,
szPassword, szDatabase );
#else
            // Compile for SNAC
            sprintf( szConnectStr,
"DRIVER=SQL Native
Client;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
szServer, szUser,
szPassword, szDatabase );
#endif
            rc = SQLDriverConnect(m_hdbc,
NULL, (SQLCHAR*)szConnectStr, sizeof(szConnectStr),
(SQLCHAR*)szOutStr,
sizeof(szOutStr), &iOutStrLen, SQL_DRIVER_NOPROMPT );

            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eConnect);
        }

        if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmt) != SQL_SUCCESS)

            ThrowError(CODBCERR::eAllocHandle);

        {
            char                buffer[128];

            // set some options affecting
connection behavior
            strcpy(buffer, "set nocount on
set XACT_ABORT ON");
            rc = SQLExecDirect(m_hstmt,
(unsigned char *)buffer, SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

            // verify that version of stored
procs on server is correct
            char db_sp_version[10];
            strcpy(buffer, "{call
tpcc_version}");

```

```

            rc = SQLExecDirect(m_hstmt,
(unsigned char *)buffer, SQL_NTS);
            if (rc != SQL_SUCCESS && rc !=
SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);
            if ( SQLBindCol(m_hstmt, 1,
SQL_C_CHAR, &db_sp_version, sizeof(db_sp_version),
NULL) != SQL_SUCCESS )

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

                ThrowError(CODBCERR::eFetch);
            if
            (strcmp(db_sp_version, sVersion))
                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION
);

            SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmt);
        }

        // Bind parameters for each of the
transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
    }

    CTPCC_ODBC::~CTPCC_ODBC( void )
    {
        // note: descriptors are automatically
released when the connection is dropped
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtNewOrder);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtPayment);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtDelivery);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtOrderStatus);
        SQLFreeHandle(SQL_HANDLE_STMT,
m_hstmtStockLevel);

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

    //void CTPCC_ODBC::ThrowError( CODBCERR::ACTION
eAction )
    void CTPCC_ODBC::ThrowError( RETCODE eAction )
    {
        RETCODE rc;
        lNativeError;
        char szState[6];
        char
        szMsg[SQL_MAX_MESSAGE_LENGTH];

```

```

char
szTmp[6*SQL_MAX_MESSAGE_LENGTH];
CODBCERR *pODBCERR;
// not allocated until needed (maybe never)

pODBCERR = new CODBCERR();

pODBCERR->m_NativeError = 0;
//pODBCERR->m_eAction = eAction;
pODBCERR->m_eAction =
(CODBCERR::ACTION)eAction;
pODBCERR->m_bDeadLock = FALSE;

szTmp[0] = 0;
szMsg[0] = 0;
while (TRUE)
{
    rc = SQLError(henv, m_hdbc,
m_hstmt, (BYTE *)&szState, &lNativeError,
(BYTE *)&szMsg, sizeof(szMsg), NULL);
    if (rc == SQL_NO_DATA)
    {
        break;
    }
    if (rc != SQL_SUCCESS)
    {
        break;
    }
    // check for deadlock
    if (lNativeError == 1205 ||
(lNativeError == iErrOleDbProvider &&
sErrTimeoutExpired) != NULL)
    pODBCERR->m_bDeadLock =
TRUE;

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

    // quit if there isn't enough
room to concatenate error text
    if ( (strlen(szMsg) + 2) >
(sizeof(szTmp) - strlen(szTmp)) )
        break;

    // include line break after first
error msg
    if (szTmp[0] != 0)
        strcat( szTmp, "\n");
    strcat( szTmp, szMsg );
}

if (pODBCERR->m_odbcerrstr != NULL)
{
    delete [] pODBCERR->m_odbcerrstr;
    pODBCERR->m_odbcerrstr = NULL;
}

```

```

}
if (strlen(szTmp) > 0)
{
    pODBCERR->m_odbcerrstr = new
char[ strlen(szTmp)+1 ];
    strcpy( pODBCERR->m_odbcerrstr,
szTmp );
}

SQLFreeStmt(m_hstmt, SQL_CLOSE);
throw pODBCERR;
}

void CTPCC_ODBC::InitStockLevelParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtStockLevel) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtStockLevel;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.StockLevel.w_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
|| SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
)
        ThrowError(CODBCERR::eBindParam);

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

    //Compose Stock Level statement
    _snprintf(m_szStockLevelCommand,
sizeof(m_szStockLevelCommand)/sizeof(m_szStockLevelCo
mmand[0]),
        L"call %stpcck_stocklevel
(?,?,?)", m_szSPPrefix);
}

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

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)
    {
        try
        {

```

```

rc =
SQLExecDirectW(m_hstmt, m_szStockLevelCommand,
SQL_NTS);
        if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

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

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        m_txn.StockLevel.exec_status_code = eOK;
        break;
    }
    catch (CODBCERR *e)
    {
        if (!e->m_bDeadLock)
            if (++iTryCount > iMaxRetries)
                throw;

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

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

void CTPCC_ODBC::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtNewOrder) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrderNoDuplicates) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols1) != SQL_SUCCESS
||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols2) != SQL_SUCCESS
)
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;
}

```



```

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

        ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindCol);

        // associate the column bindings for the
second result set
        if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.w_tax, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.NewOrder.o_id, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_last,
sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.c_discount, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_credit,
sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0,
NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindCol);

        //Compose the New Order statement
        snprintf(m_szNewOrderNoDuplicatesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNe
wOrderNoDuplicatesCommand[0]),
        L"(call
%stpc_neworder_new(?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?,?,"
        L"?,?,?,?,?,?,?,?,?,?,?,?,?,
,?,?,?)", m_szSPPrefix);

        m_iBeginNewOrderNoDuplicatesVariablePart =
33 + wcslen(m_szSPPrefix); // fixed part + prefix
part
    }

    //
    // Returns true if there are duplicate
(warehouse_id, item_id)
// lineitem pairs in New Order input
parameters.
//
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
    int i, j;

    for (i = 0; i < m_txn.NewOrder.o_ol_cnt;
++i)
        {
            for (j = i+1; j<
m_txn.NewOrder.o_ol_cnt; ++j)
                {
                    if
(m_txn.NewOrder.OL[i].ol_i_id ==
m_txn.NewOrder.OL[j].ol_i_id)
                        return true;
                }
            return false;
        }

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

```

```

        else
        {
            NewOrderDuplicates();
        }
    }

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

0 1 2 //
//
012345678901234567890123456789 //
wchar_t
szSqlTemplate[IMAX_SP_NAME_LEN];

tpcc_neworder(?,?,?,?,," // L"(call
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?," //
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)
            throw;
        // hit deadlock;
backoff for increasingly longer period
        delete e;
        Sleep(10 * iTryCount);
    }
}

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

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

    0      1      2      3                                //
//
0123456789012345678901234567890123
    wchar_t
szSqlTemplate[IMAX_SP_NAME_LEN];

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

```

```

L"?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?," //
//
L"?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?," //
        m_hstmt = m_hstmtNewOrderNoDuplicates;

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

            ThrowError(CODBCERR::eSetStmtAttr);

        // clip statement buffer based on number of
parameters
        // fixed part is 33 chars and variable part
is 6 chars per line item
        wcsncpy(szSqlTemplate,
m_szNewOrderNoDuplicatesCommand);
        i =
m_iBeginNewOrderNoDuplicatesVariablePart +
m_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::eSetStmtAttr);

                rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
                if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)

```

```

        ThrowError(CODBCERR::eExecDirect);

        // configure block
        cursor
            if
                (SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
                (SQLPOINTER)MAX_OL_NEW_ORDER_ITEMS, 0) !=
                SQL_SUCCESS)
                    ThrowError(CODBCERR::eSetStmtAttr);

        // Get order line
        results
            if ( SQLFetch(m_hstmt)
            == SQL_ERROR)
                ThrowError(CODBCERR::eFetch);

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

        // associate the column
        bindings for the second result set
            if ( SQLSetStmtAttrW(
            m_hstmt, SQL_ATTR_APP_ROW_DESC,
            m_descNewOrderNoDuplicatesCols2, SQL_IS_POINTER ) !=
            SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

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

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

        SQLFreeStmt(m_hstmt,
        SQL_CLOSE);

        // Check Fetch return
        code for no rows returned.
        // It means customer id
        or warehouse id were invalid.
        //
        if (rc == SQL_NO_DATA)

```

```

            throw new
            CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_INVALID_NEW_ORDER_
            PARAM);

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

                m_txn.NewOrder.exec_status_code = eOK;
            }
            else
                m_txn.NewOrder.exec_status_code =
                eInvalidItem;

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

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

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

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

    m_hstmt = m_hstmtPayment;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
    &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
    &m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6, 2,
    &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
    &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS

```

```

        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
    &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
    &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
    SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
    sizeof(m_txn.Payment.c_last), 0,
    &m_txn.Payment.c_last, sizeof(m_txn.Payment.c_last),
    NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

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

```

```

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

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

```

```

}

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

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

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

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

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            if (m_txn.Payment.c_id
== 0)
                throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_CUST );
            else
                m_txn.Payment.exec_status_code = eOK;

            break;
        }
        catch (CODBCERR *e)
        {
            if (!e->m_bDeadLock)
                throw;

            // hit deadlock;
            delete e;
            Sleep(10 * iTryCount);
        }
    }

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

void CTPCC_ODBC::InitOrderStatusParams()
{

```

```

        if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtOrderStatus) != SQL_SUCCESS
        ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
        ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtOrderStatus;

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

        int i = 0;
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0, 0,
&m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.OrderStatus.c_last), 0,
&m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);

        // configure block cursor
        if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) !=
SQL_SUCCESS
        || SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS
        )
            ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_supply_w_id,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_i_id, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.OL[0].ol_quantity,
0, NULL) != SQL_SUCCESS

```

```

        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

    if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

    i = 0;
    if ( SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0, NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_firSt,
sizeof(m_txn.OrderStatus.c_first), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_middle,
sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.o_entry_d,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.o_carrier_id, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.c_balance, 0, NULL)
!= SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.o_id, 0, NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

    //Compose Order Status statement
    _snprintf(m_szOrderStatusCommand,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOrderStatus
Command[0]),
        L"[call %stpc_orderstatus
(?,?,?,?)]", m_szSPPrefix);
}

void CTPCC_ODBC::OrderStatus()
{
    int
        iTryCount = 0;

    RETCODE
    rc;

```

```

    m_hstmt = m_hstmtOrderStatus;

    if (m_txn.OrderStatus.c_id != 0)
        m_txn.OrderStatus.c_last[0] = 0;

    while (TRUE)
    {
        try
        {
            if ( SQLSetStmtAttrW(
m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols1, SQL_IS_POINTER ) !=
SQL_SUCCESS )

                ThrowError(CODBCERR::eSetStmtAttr);

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

                    ThrowError(CODBCERR::eSetStmtAttr);

            rc =
SQLExecDirectW(m_hstmt, m_szOrderStatusCommand,
SQL_NTS);
            if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

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

                    ThrowError(CODBCERR::eSetStmtAttr);

            rc = SQLFetchScroll(
m_hstmt, SQL_FETCH_NEXT, 0 );
            //
            if ( !(rc ==
SQL_SUCCESS) || ((rc == SQL_SUCCESS_WITH_INFO) &&
(m_RowsFetched != 0))) )
                if ( rc !=
SQL_SUCCESS )

                    ThrowError(CODBCERR::eFetchScroll);

            m_txn.OrderStatus.o_ol_cnt =
(short)m_RowsFetched;

            if
                (m_txn.OrderStatus.o_ol_cnt != 0)
                {
                    if (
SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

```

```

                ThrowError(CODBCERR::eSetStmtAttr);

                //
                if (
SQLMoreResults(m_hstmt) == SQL_ERROR )
                    if ( rc =
SQLMoreResults(m_hstmt) != SQL_SUCCESS )
                        {
                            ThrowError(CODBCERR::eMoreResults);
                        }

                //
                if ( rc =
SQLFetch(m_hstmt) == SQL_ERROR)
                    if ( rc =
SQLFetch(m_hstmt) != SQL_SUCCESS)

                        ThrowError(CODBCERR::eFetch);

                SQLFreeStmt(m_hstmt,
SQL_CLOSE);

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

                    m_txn.OrderStatus.exec_status_code = eOK;

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

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

            if (iTryCount)
                throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
        }

        void CTPCC_ODBC::InitDeliveryParams()
        {
            if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtDelivery) != SQL_SUCCESS )

```

```

ThrowError(CODBCERR::eAllocHandle);

m_hstmt = m_hstmtDelivery;

int i = 0;
if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0, 0,
&m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindParam);

for (i=0;i<10;i++)
{
    if ( SQLBindCol(m_hstmt,
(UWORD)(i+1), SQL_C_SLONG, &m_txn.Delivery.o_id[i],
0, NULL) != SQL_SUCCESS )

        ThrowError(CODBCERR::eBindCol);
}

//Compose Delivery statement
snprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryComman
d[0]),
        L"call %stpcc_delivery (?,?)",
m_szSPPrefix);
}

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

    m_hstmt = m_hstmtDelivery;

    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt, m_szDeliveryCommand,
SQL_NTS);
            if (rc != SQL_SUCCESS
&& rc != SQL_SUCCESS_WITH_INFO)

                ThrowError(CODBCERR::eExecDirect);

            if ( SQLFetch(m_hstmt)
== SQL_ERROR )

                ThrowError(CODBCERR::eFetch);

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            m_txn.Delivery.exec_status_code = eOK;
            break;
        }
    }
}

```

```

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

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

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

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

// need to declare functions for import, unless
define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

#define IMAX_SP_NAME_LEN 256 //maximum length of a
stored procedure name with parameters

class CODBCERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eAllocConn,
        // error from SQLAllocConnect
        eAllocHandle,
        // 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."

            ERR_INVALID_NEW_ORDER_PARAM // "New Order
parameter invalid."
        };

        CTPCC_ODBC_ERR( int iErr ) {
m_errno = iErr; m_iTryCount = 0; };

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

        int m_errno;
        int m_iTryCount;

        int ErrorType()
{return ERR_TYPE_TPCC_ODBC;};
        char* ErrorTypeStr() { return
"TPCC ODBC"; }
        int ErrorNum()
{return m_errno;};

        char* ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
    private:
        // declare variables and private
functions here...
        BOOL m_bDeadlock;
// transaction was selected as
deadlock victim
        int m_MaxRetries; // retry
count on deadlock

        SQLHENV m_henv;
// ODBC environment
handle
        SQLHDBC m_hdbc;
        SQLHSTMT m_hstmt;
// the current hstmt

        SQLHSTMT m_hstmtNewOrder;
        SQLHSTMT
m_hstmtNewOrderNoDuplicates; // NewOrder
with one result set for lineitem details

```

```

        SQLHSTMT m_hstmtPayment;
        SQLHSTMT m_hstmtDelivery;
        SQLHSTMT m_hstmtOrderStatus;
        SQLHSTMT m_hstmtStockLevel;

        SQLHDESC m_descNewOrderCols1;
        SQLHDESC m_descNewOrderCols2;
        SQLHDESC
m_descNewOrderNoDuplicatesCols1; //
NewOrder with one result set for lineitem details
        SQLHDESC
m_descNewOrderNoDuplicatesCols2; //
NewOrder with one result set for lineitem details
        SQLHDESC m_descOrderStatusCols1;
        SQLHDESC m_descOrderStatusCols2;

        wchar_t
m_szSPPrefix[32]; // stored procedures
prefix

        wchar_t
m_szNewOrderCommand[IMAX_SP_NAME_LEN];
        wchar_t
m_szNewOrderNoDuplicatesCommand[IMAX_SP_NAME
E_LEN];

        int
m_iBeginNewOrderVariablePart; // beginning
of the variable part in NewOrder statement
        int
m_iBeginNewOrderNoDuplicatesVariablePart;
// beginning of the variable part in
NewOrder statement
        wchar_t
m_szPaymentCommand[IMAX_SP_NAME_LEN];
        wchar_t
m_szDeliveryCommand[IMAX_SP_NAME_LEN];
        wchar_t
m_szOrderStatusCommand[IMAX_SP_NAME_LEN];
        wchar_t
m_szStockLevelCommand[IMAX_SP_NAME_LEN];

        // new-order specific fields
        SQLINTEGER m_BindOffset;
        SQLINTEGER
m_RowsFetched;
        int
m_no_commit_flag;

        // tpcc_neworder_new flag
        BOOL
m_bCallNoDuplicatesNewOrder;

        //void ThrowError(
COBCEERR::ACTION eAction );
        void ThrowError( RETCODE eAction
);

        void InitNewOrderParams();
        void InitPaymentParams();
        void InitDeliveryParams();
        void InitStockLevelParams();
        void InitOrderStatusParams();

```

```

        union
        {
            NEW_ORDER_DATA
NewOrder;
            PAYMENT_DATA
Payment;
            DELIVERY_DATA
Delivery;
            STOCK_LEVEL_DATA
StockLevel;
            ORDER_STATUS_DATA
OrderStatus;
        }
        m_txn;

        bool DuplicatesInNewOrder();
        void NewOrderDuplicates();
        void NewOrderNoDuplicates();

    public:
        CTPCC_ODBC( LPCWSTR
szServer, LPCWSTR szUser, LPCWSTR szPassword,
LPCWSTR szHost, LPCWSTR 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
( LPCWSTR szServer, LPCWSTR szUser,
LPCWSTR szPassword,
LPCWSTR szHost, LPCWSTR szDatabase,
LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder );

```



```
typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC) (LPCSTR, LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCWSTR, BOOL);
```

tpcc_oledb.cpp

```
/* FILE: TPC_C_OLEDB.CPP
 * Microsoft
 * TPC-C Kit Ver. 4.42.000
 * Copyright
 * Microsoft, 2004
 * Written by
 * Sergey Vasilevskiy
 * All Rights Reserved
 *
 * PURPOSE: Implements OLEDB calls for TPC-C
 * Txns.
 * Contact: Charles Levine
 * (clevine@microsoft.com)
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>
#include <stddef.h>

#define DBINITCONSTANTS
#include <oledb.h>
// #include <sqloledb.h> // Use MDAC
#include <sqlncli.h> // Use SNAC
#include <oledberr.h>

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

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

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

#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[] = "4.20.000";

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

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

```
// this needs to be the same as the max length of
machine/database/user/password in Benchcraft
(engstat.h)
const static int iMaxNameLen = 32;

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

        case DLL_PROCESS_DETACH:
            break;

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

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
 *
 */

char* CTPCC_OLEDB_ERR::ErrorText(void)
{
    int i;
    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
        "Wrong version of stored procs on database
server" },
        { ERR_INVALID_CUST,
        "Invalid Customer id,name." },
        { ERR_NO_SUCH_ORDER,
        "No orders found for customer." },
        { ERR_RETRIED_TRANS,
        "Retries before transaction succeeded." },
        { 0, "" }
    };

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

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno ==
errorMsgs[i].iError )
            break;
```

```

    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dlllexport) CTPCC_OLEDB* CTPCC_OLEDB_new(
LPCSTR szServer, // name of
SQL server
LPCSTR szUser, //
user name for login
LPCSTR szPassword, // password
for login
LPCSTR szHost, //
not used
LPCSTR szDatabase, // name of
database to use
LPCWSTR szSPPrefix ) //
prefix to append to the stored procedure names
{
    return new CTPCC_OLEDB( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
LPCSTR szServer,
// name of SQL server
LPCSTR szUser,
// user name for login
LPCSTR szPassword,
// password for login
LPCSTR szHost,
// not used
LPCSTR szDatabase,
// name of database to use
LPCWSTR szSPPrefix
// prefix to append to the stored procedure
names
)
: m_pIMalloc(NULL)
{
    int
iRc;
int
i;
HRESULT hr;

IDBInitialize*
pIDBInitialize = NULL; //
data source interface
IDBProperties*
pIDBProperties = NULL;
ICommandText*
pICommandText;
// SQL command without parameters
wchar_t
szwServer[iMaxNameLen]; //
Unicode string used to convert to BSTR
```

```

        wchar_t
        szwDatabase[iMaxNameLen];    // Unicode
string used to convert to BSTR
        wchar_t
        szwUser[iMaxNameLen];        //
Unicode string used to convert to BSTR
        wchar_t
        szwPassword[iMaxNameLen];    // Unicode
string used to convert to BSTR

        // Copy stored procedures prefix
        wcsncpy(m_szsppPrefix, szsppPrefix,
sizeof(m_szsppPrefix)/sizeof(m_szsppPrefix[0]));

        // Convert single byte ANSI strings to
Unicode (for later conversion to BSTR)
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szServer, (int)strlen(szServer)+1,
szwServer, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szDatabase,
(int)strlen(szDatabase)+1, szwDatabase, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szUser, (int)strlen(szUser)+1,
szwUser, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP,
MB_PRECOMPOSED, szPassword,
(int)strlen(szPassword)+1, szwPassword, iMaxNameLen);

        // Initialize COM library to be able to use
OLE-DB interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB
component
        //hr = CoCreateInstance(CLSID_SQLOLEDB, //
GUID of SQLOLEDB component
        // Compile for SNAC
        hr = CoCreateInstance(CLSID_SQLNCLI,    //
GUID of SQLNCLI component
        NULL,
        // not defining an aggregate
component, so NULL
        CLSCTX_INPROC_SERVER,    //
run the component in our process
        IID_IDBInitialize,
        (void **) &pIDBInitialize);
        /*
        Initialize the property values needed
to establish the connection.
        */
        for(i = 0; i < 4; i++)
            VariantInit(&m_InitProperties[i].vValue);
        //Server name.
        m_InitProperties[0].dwPropertyID =
DBPROP_INIT_DATASOURCE;
        m_InitProperties[0].vValue.vt = VT_BSTR;
        m_InitProperties[0].vValue.bstrVal=
SysAllocString(szwServer);
        m_InitProperties[0].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[0].colid = DB_NULLID;
        //Database.

```

```

        m_InitProperties[1].dwPropertyID =
DBPROP_INIT_CATALOG;
        m_InitProperties[1].vValue.vt = VT_BSTR;
        m_InitProperties[1].vValue.bstrVal=
SysAllocString(szwDatabase);
        m_InitProperties[1].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[1].colid = DB_NULLID;
        //Username (login).
        m_InitProperties[2].dwPropertyID =
DBPROP_AUTH_USERID;
        m_InitProperties[2].vValue.vt = VT_BSTR;
        m_InitProperties[2].vValue.bstrVal=
SysAllocString(szwUser);
        m_InitProperties[2].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[2].colid = DB_NULLID;
        //Password.
        m_InitProperties[3].dwPropertyID =
DBPROP_AUTH_PASSWORD;
        m_InitProperties[3].vValue.vt = VT_BSTR;
        m_InitProperties[3].vValue.bstrVal=
SysAllocString(szwPassword);
        m_InitProperties[3].dwOptions =
DBPROPOPTIONS_REQUIRED;
        m_InitProperties[3].colid = DB_NULLID;
        /*
        Construct the DBPROPSET
structure (m_rgInitPropSet). The
DBPROPSET structure is used to pass an array of
DBPROP
        structures (m_InitProperties) to the
SetProperties method.
        */
        m_rgInitPropSet.guidPropertySet =
DBPROPSET_DBINIT;
        m_rgInitPropSet.cProperties = 4;
        m_rgInitPropSet.rgProperties =
m_InitProperties;
        //Set initialization properties.
        if (FAILED(hr = pIDBInitialize-
>QueryInterface(IID_IDBProperties,
        (void **) &pIDBProperties)))
        {
            ThrowError(pIDBInitialize,
COLEDBERR::eQueryInterface, "CTPCC_OLEDB()");
        }

        hr = pIDBProperties->SetProperties(1,
&m_rgInitPropSet);

        pIDBProperties->Release();
        //Now establish the connection to the data
source.
        hr = pIDBInitialize->Initialize();

        // Free BSTR property strings
        for(i = 0; i < 4; i++)
        {

```

```

SysFreeString(m_InitProperties[i].vValue.bstrVal);
        }

        hr = pIDBInitialize-
>QueryInterface(IID_IDBCreateSession, (void
**) &m_pIDBCreateSession);

        // Releasing this has no effect on the SQL
Server connection
        // of the data source object because of the
reference maintained by
        // m_pIDBCreateSession.
        pIDBInitialize->Release();
        pIDBInitialize = NULL;

        hr = m_pIDBCreateSession-
>CreateSession(NULL, IID_IDBCreateCommand, (IUnknown
**) &m_pIDBCreateCommand);
        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateSession,
COLEDBERR::eCreateSession, "CTPCC_OLEDB()");
        }

        hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID_ICommandText, (IUnknown
**) &pICommandText);
        if (FAILED(hr))
        {
            ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CTPCC_OLEDB()");
        }

        hr = pICommandText-
>SetCommandText(DBGUID_SQL, L"set nocount on set
XACT_ABORT ON");
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CTPCC_OLEDB()");
        }

        hr = pICommandText->Execute(NULL, IID_NULL,
NULL, NULL, NULL);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eExecute, "CTPCC_OLEDB()");
        }

        pICommandText->Release();

        // verify that version of stored procs on
server is correct
        CheckSPVersion();

        // Get IMalloc interface
        hr = CoGetMalloc(1, (LPMALLOC
**) &m_pIMalloc);

```

```

        // Bind parameters for each of the
transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
    }

CTPCC_OLEDB::~CTPCC_OLEDB( void )
{
    if ( m_pIMalloc != NULL)
    {
        m_pIMalloc->Release();
    }
    m_pIPaymentCommand->Release();
    m_pIDBCreateCommand->Release();
    m_pIDBCreateSession->Release();

    CoUninitialize(); // uninitialized COM
}

library
{
/*
 *      Check stored procedures version on the
server.
 */
void CTPCC_OLEDB::CheckSPVersion()
{
    HRESULT                hr;
    char
    db_sp_version[10];
    ICommandText*         pICommandText;
    IAccessor*            pIAccessor;
    IRowset*              pRowset;
    const ULONG           nOutputParams
= 1;
    // output 1st result set columns
    HACCESSOR
    hTpccVersionOutputAccessor;
    // Structure to bind in accessor
    DBBINDING
    acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
    acOutputDBBindStatus[nOutputParams];
    LONG                  cRows = 1;
    // number of rows returned in the rowset
    ULONG
    cRowsObtained;
    HROW                  rghRow;
    //returned row handles
    HROW*                 prghRow =
&rghRow;

    hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID_ICommandText, (IUnknown
**) &pICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand, "CheckSPVersion()");
    }
}

```

```

        hr = pICommandText-
>SetCommandText(DBGUID_SQL, L"{call tpcc_version}");
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eSetCommandText, "CheckSPVersion()");
        }

        hr = pICommandText-
>QueryInterface(IID_IAccessor, (void **) &pIAccessor);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eQueryInterface, "CheckSPVersion()");
        }

        // Now fill the binding information for
result set 1 output columns
        InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

        // Binding for a rowset
        SetBinding(&acOutputDBBinding[0], 0,
sizeof(db_sp_version), DBTYPE_STR);

        hr = pIAccessor->CreateAccessor(
            DBACCESSOR_ROWDATA,
            nOutputParams,
            acOutputDBBinding,
            sizeof(db_sp_version),
            &hTpccVersionOutputAccessor,
            acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "CheckSPVersion()");
        }

        hr = pICommandText->Execute(NULL,
IID_IRowset, NULL, NULL, (IUnknown **) &pRowset);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eExecute, "CheckSPVersion()");
        }

        // Fetch the result row handle(s)
        hr = pRowset->GetNextRows(DB_NULL_HCHAPTER,
0, cRows, &cRowsObtained, &prghRow);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eGetNextRows, "CheckSPVersion()");
        }

        // Fetch the actual row data by handle
        hr = pRowset->GetData(rghRow,
hTpccVersionOutputAccessor, &db_sp_version);
        if (FAILED(hr))
        {
            ThrowError(pICommandText,
COLEDBERR::eGetData, "CheckSPVersion()");
        }
}

```

```

    }

    // Release row(s)
    hr = pRowset->Release();

    pICommandText->Release();

    // Check the retrieved version
    if (strcmp(db_sp_version,sVersion))
        throw new
CTPCC_OLEDB_ERR(
CTPCC_OLEDB_ERR::ERR_WRONG_SP_VERSION );
}

void CTPCC_OLEDB::ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR
szLocation)
{
    HRESULT
    hr;
    //char
    szState[6];
    char
    szMsg[SQL_MAX_MESSAGE_LENGTH];
    char
    szTmp[6*SQL_MAX_MESSAGE_LENGTH];
    COLEDBERR
    *pOLEDBErr;
    //
    not allocated until needed (maybe never)
    int
    iLen;
    // Interfaces
    IErrorInfo*           pIErrorInfoAll
= NULL;
    IErrorInfo*           pIErrorInfoRecord
= NULL;
    IErrorRecords*       pIErrorRecords
= NULL;
    ISupportErrorInfo*   pISupportErrorInfo
= NULL;
    ISQLServerErrorInfo*
pISQLServerErrorInfo = NULL;
    ISQLErrorInfo*
pISQLErrorInfo = NULL;

    // Information used when cannot get custom
error object
    ERRORINFO
    BasicErrorInfo;
    BSTR
    bstrDescription;
    // Number of error records.
    ULONG                 nRecs;
    ULONG                 nRec;

    // SQL Server error information from
ISQLServerErrorInfo.
    SSERRORINFO*         pSSErrorInfo =
NULL;
    OLECHAR*              pSSErrorStrings =
NULL;

    assert(pObjectWithError != NULL);
}

```

```

pOLEDBErr = new COLEDBERR(szLocation);

pOLEDBErr->m_NativeError = 0;
pOLEDBErr->m_eAction = eAction;
pOLEDBErr->m_bDeadLock = FALSE;

szTmp[0] = 0;

// Only ask for error information if the
interface supports it.
// Note: SQLOLEDB provider supports error
interface, so this check is
// for good style only.
hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**)
&pISupportErrorInfo);
if (FAILED(hr))
{
    _snprintf(szMsg, sizeof(szMsg),
"SupportErrorInfo interface not supported (hr=0x%X)",
hr);
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
    throw pOLEDBErr;
}
/*if (FAILED(pISupportErrorInfo-
>InterfaceSupportsErrorInfo(IID_InterfaceWithError))
{
    _snprintf(szMsg, sizeof(szMsg),
"InterfaceWithError
interface not supported");
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
    return;
}*/

// Do not test the return of GetErrorInfo.
It can succeed and return
// a NULL pointer in pErrorInfoAll. Simply
test the pointer.
GetErrorInfo(0, &pErrorInfoAll);

if (pErrorInfoAll != NULL)
{
    // Test to see if it's a valid
OLE DB IErrorInfo interface
    // exposing a list of records.
    if (SUCCEEDED(pErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords))
    {
        pIErrorRecords-
>GetRecordCount(&nRecs);

        // Within each record,
retrieve information from each
        // of the defined
interfaces.

```

```

for (nRec = 0; nRec <
nRecs; nRec++)
{
    // Request
the generic SQL error interface.
    pIErrorRecords->GetCustomErrorObject(nRec,
    IID_ISQLErrorInfo, // generic SQL error
interface
    (IUnknown**) &pISQLErrorInfo);
    if
    (pISQLErrorInfo != NULL)
    {
        //
Request SQL Server-specific error interface, not the
generic SQL error interface.
        pISQLErrorInfo->QueryInterface(
    IID_ISQLServerErrorInfo, // SQL Server
error interface
    (void**) &pISQLServerErrorInfo);
    }
    // Test to
ensure the reference is valid, then
    // get error
information from ISQLServerErrorInfo.
    if
    (pISQLServerErrorInfo != NULL)
    {
        pISQLServerErrorInfo-
>GetErrorInfo(&pSSErrorInfo, &pSSErrorStrings);
        //
ISQLServerErrorInfo::GetErrorInfo succeeds
        //
even when it has nothing to return. Test the
        //
pointers before using.
        if
        (pSSErrorInfo)
        {
            // First, add the error message.
            // Convert Unicode error string to ANSI.
            WideCharToMultiByte(CP_THREAD_ACP, 0,
                pSSErrorInfo->pwszMessage, -1,
                szMsg, sizeof(szMsg),
                NULL, NULL);

```

```

// quit if there isn't enough room to
concatenate error text
    if ( (strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
        break;

// include line break after first error msg
if (szTmp[0] != 0)
    strcat( szTmp, "\r\n");

// concatenate the error record to the
overall error message
    strcat( szTmp, szMsg );

// Second, add the stored procedure name
and line number, if available.

    if (wcslen(pSSErrorInfo->pwszProcedure)>0)
    {
        // Prefix with a line break
        iLen = sprintf(szMsg,
"\r\nProcedure: ");

        // Convert Unicode error string
to ANSI.
        WideCharToMultiByte(CP_THREAD_ACP, 0,
            pSSErrorInfo-
>pwszProcedure, -1,
            &szMsg[iLen],
            sizeof(szMsg) - iLen,
            NULL, NULL);

        // Check if have space to add the
line number.
        // Assume the line number takes
no more than 3 digits.
        if ((strlen(szMsg) + 4) <
sizeof(szMsg))
    {

```

```

        _snprintf(&szMsg[strlen(szMsg)],
sizeof(szMsg),
                "%:d",
pSSErrorInfo->wLineNumber);
    }

    // quit if there isn't enough
room to concatenate error text
    if ( (strlen(szMsg) + 2) >
(sizeof(szTmp) - strlen(szTmp)) )
        break;

    // concatenate the error record
to the overall error message
    strcat( szTmp, szMsg );

    // copy the overall error string
to the exception
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szTmp)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szTmp);
}

// Third, capture the (first) database
error
    if (pOLEDBErr->m_NativeError == 0 &&
pSSErrorInfo->lNative != 0)
    {
        pOLEDBErr->m_NativeError =
pSSErrorInfo->lNative;

        // Check for deadlock error code
and set the deadlock flag
        if (pSSErrorInfo->lNative ==
1205)
        {
            pOLEDBErr->m_bDeadLock
= TRUE;
        }
    }

```

```

    }

    // IMalloc::Free needed to release
references
    // on returned values.
    if (m_pIMalloc != NULL)
    {
        m_pIMalloc-
>Free(pSSErrorStrings);
        m_pIMalloc->Free(pSSErrorInfo);
    }
}

pISQLServerErrorInfo->Release();
}
else
{
    Custom error object is not supported. //
Use general OLE-DB error interface. //
Get the numeric error code //
    pIErrorRecords->GetBasicErrorInfo(nRec,
&BasicErrorInfo);
    if
(pOLEDBErr->m_NativeError == 0)
    {
        // Get the failed call HRESULT code, which
is not really the native error
        pOLEDBErr->m_NativeError =
BasicErrorInfo.hrError;
    }
    //
Try to get the string description of the error. //
    pIErrorRecords->GetErrorInfo(nRec,
LOCALE_USER_DEFAULT,
(IErrorInfo**) &pIErrorInfoRecord);
    if
(pIErrorInfoRecord)
    {
        pIErrorInfoRecord-
>GetDescription(&bstrDescription);
    }
}

```

```

// Convert Unicode error string to ANSI.
WideCharToMultiByte(CP_THREAD_ACP, 0,
                    bstrDescription, -1,
                    szMsg, sizeof(szMsg),
                    NULL, NULL);

    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}
} // for()
} // if
(SUCCEEDED(pIErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords))
    else
    {
        // No IErrorRecords
interface supported. Use default IErrorInfo.
        // Note: SQLOLEDB
supports IErrorRecords, so this check is for good
style only.
        _snprintf(szMsg,
sizeof(szMsg), "IErrorRecords interface not
supported");
        pOLEDBErr-
>m_OLEDBErrStr = new char[strlen(szMsg)+1];
        strcpy(pOLEDBErr-
>m_OLEDBErrStr, szMsg);
    }
    pIErrorInfoAll->Release();
} // if (pIErrorInfoAll != NULL)
else
{
    // No IErrorInfo interface
supported.
    // Note: SQLOLEDB supports
IErrorInfo, so this check is for good style only.
    _snprintf(szMsg, sizeof(szMsg),
"IErrorInfo interface not supported");
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
}
    throw pOLEDBErr;
}
/*
*

```

```

*         Create a new command object from the SQL
text passed in.
*
*/
void CTPCC_OLEDB::CreateCommand(wchar_t*
szSqlCommand, // I: SQL
query for the command

                                ICommandText**
ppICommandText // O: returned command object
)
{
    HRESULT hr;

    // Create a new command object
    hr = m_pIDBCreateCommand-
>CreateCommand(NULL, IID_ICommandText, (IUnknown
**)ppICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,
COLEDBERR::eCreateCommand,
"CTPCC_OLEDB::CreateCommand");
    }

    // Set command text
    hr = (*ppICommandText)-
>SetCommandText(DBGUID_SQL, szSqlCommand);
    if (FAILED(hr))
    {
        ThrowError(*ppICommandText,
COLEDBERR::eSetCommandText,
"CTPCC_OLEDB::CreateCommand");
    }

    // Prepare the command
    PrepareCommand(*ppICommandText);
}

/*
*         QueryInterface and Prepare in one function
for simplicity.
*         DEFERRED PREPARE property is set to off to
prepare 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[nOutputParams];

    // 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_pIStockLevelCommand,
COLEDBERR::eQueryInterface,
"InitStockLevelParams()");
    }

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_PARAMETERDATA,
        nInputParams,
        acInputDBBinding,
        sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelInputAccessor,
        acInputDBBindStatus);

    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }

    m_StockLevelExecuteParams.cParamSets = 1;
    m_StockLevelExecuteParams.hAccessor =
m_hStockLevelInputAccessor;
    m_StockLevelExecuteParams.pData =
&m_txn.StockLevel;

    // Now fill the binding information for
    result set 1 output columns
    InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

    // Binding for a rowset that may return
    more than one row.
    i = 0;
    // StockLevel output column 1
    SetBinding(&acOutputDBBinding[i++],
offsetof(STOCK_LEVEL_DATA, low_stock),
sizeof(m_txn.StockLevel.low_stock), DBTYPE_I4);

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,
        sizeof(STOCK_LEVEL_DATA),

&m_hStockLevelOutputAccessor,
        acOutputDBBindStatus);

    if (FAILED(hr))
    {

```

```

        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }
}

void CTPCC_OLEDB::StockLevel()
{
    HRESULT hr;
    int 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_pIStockLevelCommand,
COLEDBERR::eExecute, "StockLevel()");
            }

            // Fetch the result row
            handle(s)
            hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
            if (FAILED(hr))
            {
                ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetNextRows, "StockLevel()");
            }

            // Fetch the actual row
            data by handle
            hr = pRowset-
>GetData(rghRow, m_hStockLevelOutputAccessor,
&m_txn.StockLevel);
            if (FAILED(hr))
            {
                ThrowError(m_pIStockLevelCommand,
COLEDBERR::eGetData, "StockLevel()");
            }
        }
    }
}

```

```

// Release row(s)
hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
// Release rowset
hr = pRowset-
>Release();

m_txn.StockLevel.exec_status_code = eOK;

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

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

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{
int i, j, iOlCount;
HRESULT hr;
wchar_t
szName[iMAX_SP_NAME_LEN];
IAccessor*
pIAccessor;
const ULONG
nInputParams = 5 +
3*MAX_OL_NEW_ORDER_ITEMS; // input parameters
const ULONG
nOutputParams = 5; // output 1st result
set columns
const ULONG
nOutputParams2 = 8; // output 2nd result
set columns
// Structure to bind in accessor
DBBINDING
acInputDBBinding[nInputParams];
DBBINDSTATUS
acInputDBBindStatus[nInputParams];
DBBINDING
acOutputDBBinding[nOutputParams];
DBBINDSTATUS
acOutputDBBindStatus[nOutputParams];
DBBINDING
acOutputDBBinding2[nOutputParams2];

DBBINDSTATUS
acOutputDBBindStatus2[nOutputParams2];

// Describe the consumer buffer by filling
in the array
// of DBBINDING structures. Each binding
associates
// a single parameter to the consumer's buffer.
InitBindings(&acInputDBBinding[0],
nInputParams, eInputParameter);

i = 0;
// NewOrder parameter 1
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, w_id),
sizeof(m_txn.NewOrder.w_id), DBTYPE_I4);

// NewOrder parameter 2
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, d_id),
sizeof(m_txn.NewOrder.d_id), DBTYPE_UI1);

// NewOrder parameter 3
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, c_id),
sizeof(m_txn.NewOrder.c_id), DBTYPE_I4);

// NewOrder parameter 4
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

// NewOrder parameter 5
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, o_all_local),
sizeof(m_txn.NewOrder.o_all_local), DBTYPE_UI1);

for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
{
SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_i_id),
sizeof(m_txn.NewOrder.OL[j].ol_i_id), DBTYPE_I4);

SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_supply_w_id),
sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id),
DBTYPE_I4);

SetBinding(&acInputDBBinding[i++],
offsetof(NEW_ORDER_DATA, OL[j].ol_quantity),
sizeof(m_txn.NewOrder.OL[j].ol_quantity), DBTYPE_I2);
}

// Now fill the binding information for
result set 1 output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

// Binding for the order line rowsets (each
consist of one row).

// Bind to offsets of the OL_NEW_ORDER_DATA
structure instead of NEW_ORDER_DATA.
// IRowset::GetData() will be passed
individual array slots OL[i] to fetch the data
// from the row set.

i = 0;
// NewOrder output column 1
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name), DBTYPE_STR);

// NewOrder output column 2
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock), DBTYPE_I2);

// NewOrder output column 3
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_brand_generic),
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic),
DBTYPE_STR);

// NewOrder output column 4
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price), DBTYPE_R8);

// NewOrder output column 5
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_NEW_ORDER_DATA, ol_amount),
sizeof(m_txn.NewOrder.OL[0].ol_amount), DBTYPE_R8);

// Now fill the binding information for
result set 2 output columns
InitBindings(&acOutputDBBinding2[0],
nOutputParams2, eOutputColumn);

i = 0;
// NewOrder output column 1
SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, w_tax),
sizeof(m_txn.NewOrder.w_tax), DBTYPE_R8);

// NewOrder output column 2
SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, d_tax),
sizeof(m_txn.NewOrder.d_tax), DBTYPE_R8);

// NewOrder output column 3
SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_id),
sizeof(m_txn.NewOrder.o_id), DBTYPE_I4);

// NewOrder output column 4
SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_last),
sizeof(m_txn.NewOrder.c_last), DBTYPE_STR);

// NewOrder output column 5

```



```

        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_discount),
sizeof(m_txn.NewOrder.c_discount), DBTYPE_R8);

        // NewOrder output column 6
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, c_credit),
sizeof(m_txn.NewOrder.c_credit), DBTYPE_STR);

        // NewOrder output column 7
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_entry_d),
sizeof(m_txn.NewOrder.o_entry_d),
DBTYPE_DBTIMESTAMP);

        // NewOrder output column 8
        SetBinding(&acOutputDBBinding2[i++],
offsetof(NEW_ORDER_DATA, o_commit_flag),
sizeof(m_txn.NewOrder.o_commit_flag), DBTYPE_I2);

        for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
            // Set command text first
            // Print the fixed first portion
            // of parameters
            i = _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),

                L"call %stpcc_neworder (?,?,?,?,"
m_szSPPrefix);

            // Now print the variable portion
            // depending on the number of order line parameters
            for (iOlCount = 0; iOlCount <= j;
++iOlCount)
            {
                i +=
                _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"?,?,");
            }

            // Print the fixed end
            if (j != MAX_OL_NEW_ORDER_ITEMS -
1)
            {
                // append 'default' for
the parameters that are not used
                i +=
                _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"default)");
            }
            else // using all 15 order
line parameters
            {
                i +=
                _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"");
            }

            // Create and Prepare a new
command object for NewOrder.

```

```

        CreateCommand(szName,
&m_pINewOrderCommand[j]);

        // Now create the input accessor
for this prepared command
        hr = m_pINewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void **)&piAccessor);
        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[j],
COLEDBERR::eQueryInterface, "InitNewOrderParams()");
        }

        hr = piAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,

3 * (j + 1),

        acInputDBBinding,

        sizeof(NEW_ORDER_DATA),

        &m_hNewOrderInputAccessor[j],

        acInputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(piAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
        }

        m_NewOrderExecuteParams[j].cParamSets = 1;
        //
m_NewOrderExecuteParams.hAccessor is set dynamically
at run-time
        // based on the number of new
order items for the particular transaction call.
        m_NewOrderExecuteParams[j].hAccessor =
m_hNewOrderInputAccessor[j];
        m_NewOrderExecuteParams[j].pData
= &m_txn.NewOrder;

        // Create accessor for the first
rowset
        hr = piAccessor->CreateAccessor(
DBACCESSOR_OPTIMIZED,

        nOutputParams,
        acOutputDBBinding,

        sizeof(OL_NEW_ORDER_DATA),

```

```

        &m_hNewOrderOutputAccessor[j],
        acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(piAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
        }

        // Create accessor for the second
rowset
        hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),

        &m_hNewOrderOutputAccessor2[j],
        acOutputDBBindStatus2);
        if (FAILED(hr))
        {
            ThrowError(piAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
        }

        piAccessor->Release();
    }

void CTPCC_OLEDB::NewOrder()
{
    HRESULT                hr;
    int                    iTryCount = 0;
    IMultipleResults*     pMultipleResults;
    IRowset*              pRowset;
    IRowset*              pRowset2;
    LONG                  cRows = 1; // number of rows
    returned in the 1st rowset
    ULONG                 cRowsObtained;
    HROW                  rghRows; //returned row handles
    for the 1st result set
    HROW*                 prghRows = &rghRows;
    LONG                  cRows2 = 1; // number of rows
    returned in the 2nd rowset
    ULONG                 cRowsObtained2;
    HROW                  rghRows2; //returned row handle
    for the 2nd result set
    HROW*                 prghRows2 = &rghRows2;
    int                    i;
    long                  lRowsAffected; // the number of
affected rows for a rowset

```

```

        int
        iHandleIndex; // index into the
handle arrays based on the orders count

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

        iHandleIndex = m_txn.NewOrder.o_ol_cnt - 1;
// for convenience

        while (TRUE)
        {
            try
            {
                // Execute the prepared
command (according to the number of new orders)
                // Ask for
IMultipleResults because it returns 2 rowsets.
                hr =
m_pINewOrderCommand[iHandleIndex]->Execute(

                NULL, IID_IMultipleResults,

                &m_NewOrderExecuteParams[iHandleIndex],

                NULL,

                (IUnknown **)&MultipleResults);
                if (FAILED(hr))
                {
                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eExecute, "NewOrder()");
                }

                // Get order line
results

                m_txn.NewOrder.total_amount = 0;
                for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; ++i)

```

```

        { // Get the
        first rowset object
                hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&lRowsAffected, (IUnknown **)&pRowset);
                if
(FAILED(hr))
                {
                    char szTmp[256];

                    _snprintf(szTmp, sizeof(szTmp), "NewOrder()
result set %d, hr=0x%X", i, hr);

                    ThrowError(m_pINewOrderCommand[m_txn.NewOrd
er.o_ol_cnt - 1], COLEDBERR::eGetResult, szTmp);
                }

                // Fetch the
result row handle(s)
                hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);
                if
(FAILED(hr))
                {
                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetNextRows, "NewOrder()");
                }

                // Fetch the
actual row data by handle
                hr = pRowset-
>GetData(rghRows,
m_hNewOrderOutputAccessor[iHandleIndex],
&m_txn.NewOrder.OL[i]);
                if
(FAILED(hr))
                {
                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetData, "NewOrder()");
                }

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

                // Release
row(s)
                hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL,
NULL);

                // Release
rowset
                hr = pRowset-
>Release();
            }
        }

```

```

                // Get the second
rowset object
                hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **)&pRowset2);
                if (FAILED(hr))
                {
                    char
szTmp[256];

                    _snprintf(szTmp, sizeof(szTmp), "NewOrder()
result set %d, hr=%d", i, hr);

                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetResult, szTmp);
                }

                // Fetch the result row
handle(s)
                hr = pRowset2-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
                if (FAILED(hr))
                {
                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetNextRows, "NewOrder()");
                }

                // Fetch the actual row
data by handle
                hr = pRowset2-
>GetData(rghRows2,
m_hNewOrderOutputAccessor2[iHandleIndex],
&m_txn.NewOrder);
                if (FAILED(hr))
                {
                    ThrowError(m_pINewOrderCommand[iHandleIndex
], COLEDBERR::eGetData, "NewOrder()");
                }

                // Release row(s)
                hr = pRowset2-
>ReleaseRows(cRowsObtained2, prghRows2, NULL, NULL,
NULL);

                // Release rowset
                hr = pRowset2-
>Release();

                // Release the common
MultipleResults interface
                hr = pMultipleResults-
>Release();

                if
(m_txn.NewOrder.o_all_local == 1)

```

```

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

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else
        {
            m_txn.NewOrder.exec_status_code =
eInvalidItem;
        }
        break;
    }
    catch (COLEDBERR *e)
    {
        if (!(e->m_bDeadLock))
|| (++iTryCount > iMaxRetries))
            throw;

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

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitPaymentParams ()
{
    int
        i;
    HRESULT
        hr;
    wchar_t
        szName[IMAX_SP_NAME_LEN];
    IAccessor*
        pIAccessor;
    const
        ULONG
        nInputParams = 7; // input parameters
        const ULONG
        nOutputParams = 27; // output result set
columns
    // Structure to bind in accessor
    DBBINDING
        acInputDBBinding[nInputParams];
    DBINDSTATUS
        acInputDBBindStatus[nInputParams];
    DBBINDING
        acOutputDBBinding[nOutputParams];
    DBINDSTATUS
        acOutputDBBindStatus[nOutputParams];

    // Set command text

```

```

        snwprintf(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);

        // Payment parameter 7
        SetBinding(&acInputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

        hr = m_pIPaymentCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
        if (FAILED(hr))
        {
            ThrowError(m_pIPaymentCommand,
COLEDBERR::eQueryInterface, "InitPaymentParams ()");
        }

        hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,

```

```

        nInputParams,
        acInputDBBinding,
        sizeof(PAYMENT_DATA),
        &m_hPaymentInputAccessor,
        acInputDBBindStatus);

        if (FAILED(hr))
        {
            ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams ()");
        }

        m_PaymentExecuteParams.cParamSets = 1;
        m_PaymentExecuteParams.hAccessor =
m_hPaymentInputAccessor;
        m_PaymentExecuteParams.pData =
&m_txn.Payment;

        // Now fill the binding information for
        output columns
        InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

        i = 0;
        // Payment output column 1
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

        // Payment output column 2
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

        // Payment output column 3
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, h_date),
sizeof(m_txn.Payment.h_date), DBTYPE_DBTIMESTAMP);

        // Payment output column 4
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_1),
sizeof(m_txn.Payment.w_street_1), DBTYPE_STR);

        // Payment output column 5
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_street_2),
sizeof(m_txn.Payment.w_street_2), DBTYPE_STR);

        // Payment output column 6
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_city),
sizeof(m_txn.Payment.w_city), DBTYPE_STR);

        // Payment output column 7
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_state),
sizeof(m_txn.Payment.w_state), DBTYPE_STR);

        // Payment output column 8
        SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, w_zip),
sizeof(m_txn.Payment.w_zip), DBTYPE_STR);

```

```

// Payment output column 9
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

// Payment output column 10
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 11
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 12
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 13
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 14
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_first),
sizeof(m_txn.Payment.c_first), DBTYPE_STR);

// Payment output column 15
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_middle),
sizeof(m_txn.Payment.c_middle), DBTYPE_STR);

// Payment output column 16
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

// Payment output column 17
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 18
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 19
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 20
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 21

```

```

SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_phone),
sizeof(m_txn.Payment.c_phone), DBTYPE_STR);

// Payment output column 22
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_since),
sizeof(m_txn.Payment.c_since), DBTYPE_DBTIMESTAMP);

// Payment output column 23
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit),
sizeof(m_txn.Payment.c_credit), DBTYPE_STR);

// Payment output column 24
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_credit_lim),
sizeof(m_txn.Payment.c_credit_lim), DBTYPE_R8);

// Payment output column 25
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_discount),
sizeof(m_txn.Payment.c_discount), DBTYPE_R8);

// Payment output column 26
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_balance),
sizeof(m_txn.Payment.c_balance), DBTYPE_R8);

// Payment output column 27
SetBinding(&acOutputDBBinding[i++],
offsetof(PAYMENT_DATA, c_data),
sizeof(m_txn.Payment.c_data), DBTYPE_STR);

hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(PAYMENT_DATA),
&m_hPaymentOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))
{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor, "InitPaymentParams()");
}

void CTPCC_OLEDB::Payment()
{
HRESULT hr;
int
iTryCount = 0;
IRowset* pRowset;
LONG cRows = 1;
// number of rows returned in the rowset
ULONG
cRowsObtained;
HROW rghRow;
//returned row handles

```

```

HROW* prghRow =
&rghRow;

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

while (TRUE)
{
try
{
// Execute the prepared
command
hr =
m_pIPaymentCommand->Execute(NULL, IID_IRowset,
&m_PaymentExecuteParams, NULL,

(IUnknown **) &pRowset);
if (FAILED(hr))
{
ThrowError(m_pIPaymentCommand,
COLEDBERR::eExecute, "Payment()");
}

// Fetch the result row
handle(s)
hr = pRowset->
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
if (FAILED(hr))
{
ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetNextRows, "Payment()");
}

// Fetch the actual row
data by handle
hr = pRowset->
>GetData(rghRow, m_hPaymentOutputAccessor,
&m_txn.Payment);
if (FAILED(hr))
{
ThrowError(m_pIPaymentCommand,
COLEDBERR::eGetData, "Payment()");
}

// Release row(s)
hr = pRowset->
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
// Release rowset
hr = pRowset->
>Release();

if (m_txn.Payment.c_id
== 0)
throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
else

```

```

        m_txn.Payment.exec_status_code = eOK;
        break;
    }
    catch (COLEDBERR *e)
    {
        if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
            throw;

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

// if (iTryCount)
//     throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitOrderStatusParams()
{
    int
        i;
    HRESULT
        hr;
    wchar_t
        szName[IMAX_SP_NAME_LEN];
    IAccessor*
        pIAccessor;
    const ULONG
        nInputParams = 4; // input parameters
        nOutputParams = 5; // output 1st result
set columns
        const ULONG
        nOutputParams2 = 8; // output 2nd result
set columns
// Structure to bind in accessor
DBBINDING
acInputDBBinding[nInputParams];
DBBINDSTATUS
acInputDBBindStatus[nInputParams];
DBBINDING
acOutputDBBinding[nOutputParams];
DBBINDSTATUS
acOutputDBBindStatus[nOutputParams];
DBBINDING
acOutputDBBinding2[nOutputParams2];
DBBINDSTATUS
acOutputDBBindStatus2[nOutputParams2];

// Set command text
_snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"call
%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_pIOrderStatusCommand,
COLEDBERR::eQueryInterface,
"InitOrderStatusParams()");
}

hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(ORDER_STATUS_DATA),
&m_hOrderStatusInputAccessor,
acInputDBBindStatus);

if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

m_OrderStatusExecuteParams.cParamSets = 1;
m_OrderStatusExecuteParams.hAccessor =
m_hOrderStatusInputAccessor;

```

```

        m_OrderStatusExecuteParams.pData =
&m_txn.OrderStatus;

// Now fill the binding information for
result set 1 output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

// Binding for a rowset that may return
more than one row.
// Bind to offsets of the
OL_ORDER_STATUS_DATA structure instead of
ORDER_STATUS_DATA.
// IRowset::GetData() will be passed
individual array slots OL[i] to fetch the data
// from the row set.

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_supply_w_id),
sizeof(m_txn.OrderStatus.OL[0].ol_supply_w_id),
DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_txn.OrderStatus.OL[0].ol_i_id),
DBTYPE_I4);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_quantity),
sizeof(m_txn.OrderStatus.OL[0].ol_quantity),
DBTYPE_I2);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_amount),
sizeof(m_txn.OrderStatus.OL[0].ol_amount),
DBTYPE_R8);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_delivery_d),
sizeof(m_txn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_ORDER_STATUS_DATA),
&m_hOrderStatusOutputAccessor,
acOutputDBBindStatus);

if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

```

```

// Now fill the binding information for
result set 2 output columns
InitBindings(&acOutputDBBinding2[0],
nOutputParams2, eOutputColumn);

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_txn.OrderStatus.c_first), DBTYPE_STR);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_txn.OrderStatus.c_middle), DBTYPE_STR);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_entry_d),
sizeof(m_txn.OrderStatus.o_entry_d),
DBTYPE_DBTIMESTAMP);

// OrderStatus output column 7
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_carrier_id),
sizeof(m_txn.OrderStatus.o_carrier_id), DBTYPE_I2);

// OrderStatus output column 8
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_balance),
sizeof(m_txn.OrderStatus.c_balance), DBTYPE_R8);

// OrderStatus output column 9
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_id),
sizeof(m_txn.OrderStatus.o_id), DBTYPE_I4);

hr = piAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, //
cannot be optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),

&m_hOrderStatusOutputAccessor2,
acOutputDBBindStatus2);

if (FAILED(hr))
{
ThrowError(piAccessor,
COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
}

```

```

}

void CTPCC_OLEDB::OrderStatus()
{
HRESULT hr;
int
iTryCount = 0;
IMultipleResults* pMultipleResults;
IRowset* pRowset;
IRowset* pRowset2;
LONG
cRows = MAX_OL_ORDER_STATUS_ITEMS; //
number of rows returned in the 1st rowset
ULONG
cRowsObtained;
HROW
rghRows[MAX_OL_ORDER_STATUS_ITEMS];
//returned row handles for the 1st result
set
HROW*
prghRows = &rghRows[0];
LONG
cRows2 = 1; // number of rows
returned in the 2nd rowset
ULONG
cRowsObtained2;
HROW
rghRows2; //returned row handle
for the 2nd result set
HROW*
prghRows2 = &rghRows2;
int
i;
long
lRowsAffected; // the number of
affected rows for a rowset

if (m_txn.OrderStatus.c_id != 0)
m_txn.OrderStatus.c_last[0] = 0;

while (TRUE)
{
try
{
// Execute the prepared
command // Ask for
IMultipleResults because it returns 2 rowsets.
hr =
m_piOrderStatusCommand->Execute(NULL,
IID_IMultipleResults, &m_OrderStatusExecuteParams,
NULL,

(IUnknown **)&pMultipleResults);
if (FAILED(hr))
{
ThrowError(m_piOrderStatusCommand,
COLEDBERR::eExecute, "OrderStatus()");
}
}
}

```

```

////////////////////////////////////
// Get order line
results
////////////////////////////////////

// Get the first rowset
object
hr = pMultipleResults-
>GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **)&pRowset);
if (FAILED(hr))
{
ThrowError(m_piOrderStatusCommand,
COLEDBERR::eGetResult, "OrderStatus()");
}

// Fetch the result row
handle(s)
hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);
if (FAILED(hr))
{
ThrowError(m_piOrderStatusCommand,
COLEDBERR::eGetNextRows, "OrderStatus()");
}

m_txn.OrderStatus.o_ol_cnt =
(short)cRowsObtained;

// Get the data from
multiple rows in this rowset
for (i = 0; i <
m_txn.OrderStatus.o_ol_cnt; ++i)
{
// Fetch the
actual row data by handle
hr = pRowset-
>GetData(rghRows[i], m_hOrderStatusOutputAccessor,
&m_txn.OrderStatus.OL[i]);
if
(FAILED(hr))
{
ThrowError(m_piOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
}
}

// Release row(s)
hr = pRowset-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL,
NULL);

// Release rowset
hr = pRowset-
>Release();

```

```

////////////////////////////////////
// Get the second
rowset object

////////////////////////////////////
if
(m_txn.OrderStatus.o_ol_cnt > 0)
{
    hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&lRowsAffected, (IUnknown **)&pRowset2);
    if
(FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetResult, "OrderStatus()");
    }
    // Fetch the
result row handle(s)
    hr =
pRowset2->GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);
    if
(FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetNextRows, "OrderStatus()");
    }
    // Fetch the
actual row data by handle
    hr =
pRowset2->GetData(rghRows2,
m_hOrderStatusOutputAccessor2, &m_txn.OrderStatus);
    if
(FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
    }
    // Release
row(s)
    hr =
pRowset2->Release();
}
// Release the common
MultipleResults interface
hr = pMultipleResults-
>Release();

if
(m_txn.OrderStatus.o_ol_cnt == 0)
    throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_NO_SUCH_ORDER
);

```

```

else if
(m_txn.OrderStatus.c_id == 0 &&
m_txn.OrderStatus.c_last[0] == 0)
    throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_INVALID_CUST );
else
    m_txn.OrderStatus.exec_status_code = eOK;
    break;
}
catch (COLEDBERR *e)
{
    if (!(e->m_bDeadLock)
|| (++iTryCount > iMaxRetries))
        throw;
    // hit deadlock;
backoff for increasingly longer period
    delete e;
    Sleep(10 * iTryCount);
}
}
// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_OLEDB::InitDeliveryParams()
{
    int i;
    HRESULT hr;
    wchar_t
szName[IMAX_SP_NAME_LEN];
    IAccessor*
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[nOutputParams];

    // 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.cParamSets = 1;
m_DeliveryExecuteParams.hAccessor =
m_hDeliveryInputAccessor;
m_DeliveryExecuteParams.pData =
&m_txn.Delivery;

// Now fill the binding information for
result set 1 output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

// Binding for a rowset that may return
more than one row.
for (i = 0; i < 10; ++i)
{
    // Delivery output column 1

```

```

        SetBinding(&acOutputDBBinding[i],
offsetof(DELIVERY_DATA, o_id[i]),
sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
    }

    hr = piAccessor->CreateAccessor(
        DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,
        sizeof(DELIVERY_DATA),
&m_hDeliveryOutputAccessor,
        acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(piAccessor,
COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
    }
}

void CTPCC_OLEDB::Delivery()
{
    HRESULT          hr;
    int              iTryCount = 0;
    IRowset*         pRowset;
    LONG             cRows = 1;
    // number of rows returned in the rowset
    ULONG           cRowsObtained;
    HROW            rgRow;
    //returned row handles
    HROW*           prghRow =
&rgRow;

    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(rgRow, m_hDeliveryOutputAccessor,
&m_txn.Delivery);
            if (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetData, "Delivery()");
            }
            // Release row(s)
            hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);
            // Release rowset
            hr = pRowset-
>Release();

            m_txn.Delivery.exec_status_code = eOK;
            break;
        }
        catch (COLEDBERR *e)
        {
            if (!(e->m_bDeadLock))
            {
                ++iTryCount > iMaxRetries)
                throw;
            }
            // hit deadlock;
            // backoff for increasingly longer period
            delete e;
            Sleep(10 * iTryCount);
        }
    }

    // if (iTryCount)
    // throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

tpcc_oledb.h
/* FILE: TPC_C_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 DllDecl
#define DllDecl __declspec( dllimport )
#endif

#define iMAX_SP_NAME_LEN 256 //maximum length of a
stored procedure name with parameters

// Type of parameter and result set column bindings.
enum eBindingType
{
    eInputParameter,
    eOutputParameter,
    eInputOutputParameter,
    eOutputColumn
};

class COLEDBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eQueryInterface,
        // error from QueryInterface
        eCreateSession,
        eCreateCommand,
        eSetCommandText,
        eExecute,
        // = 6
        eCreateAccessor,
        ePrepare,
        eGetNextRows,
        eGetData,
        eGetResult
        // = 11
    };

    COLEDBERR(LPCTSTR szLoc)
        : CBaseErr(szLoc)
    {
        m_eAction = eNone;
        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_OLEDBErrStr = NULL;
    };

    ~COLEDBERR()
    {
        if (m_OLEDBErrStr !=
NULL)
            delete []
m_OLEDBErrStr;
    }
};

```



```

    };
    ACTION    m_eAction;
    int
    m_NativeError;
    BOOL      m_bDeadLock;
    char      *m_OLEDBErrStr;

    int
    ErrorType()
    {return ERR_TYPE_OLEDB;};
    char*
    ErrorTypeStr() { return
    "OLEDB"; }
    int
    ErrorNum()
    {return m_NativeError;};
    char*
    ErrorText() {return
    m_OLEDBErrStr;};
    int
    ErrorAction()
    { return (int)m_eAction; }
};

class CTPCC_OLEDB_ERR : public CBaseErr
{
public:
    enum TPCC_OLEDB_ERRS
    {
        ERR_WRONG_SP_VERSION =
1, // "Wrong version of stored procs on
database server"
        ERR_INVALID_CUST,
// "Invalid Customer id,name."
        ERR_NO_SUCH_ORDER,
// "No orders found for
customer."
        ERR_RETRIED_TRANS,
// "Retries before transaction
succeeded."
    };
    CTPCC_OLEDB_ERR( int iErr ) {
m_errno = iErr; m_iTryCount = 0; };
    CTPCC_OLEDB_ERR( int iErr, int
iTryCount ) { m_errno = iErr; m_iTryCount =
iTryCount; };

    int
    m_errno;
    int
    m_iTryCount;

    int
    ErrorType()
    {return ERR_TYPE_TPCC_OLEDB;};
    char*
    ErrorTypeStr() { return
    "TPCC OLEDB"; }
    int
    ErrorNum()
    {return m_errno;};

    char*
    ErrorText();
};

class DllDecl CTPCC_OLEDB : public CTPCC_BASE
{
private:

```

```

// declare variables and private
functions here...
    BOOL
    m_bDeadlock; //
transaction was selected as deadlock victim
    int
    m_MaxRetries;
// retry count on deadlock

    DBPROPSET
    m_rgInitPropSet; //
initialization property set used to establish a
connection
    DBPROP
    m_InitProperties[4]; //
individual initialization properties
    IDBCreateSession*
    m_pIDBCreateSession; // session
(connection) interface
    IDBCreateCommand*
    m_pIDBCreateCommand; // SQL
command creation interface

    IMalloc*
    m_pIMalloc;
// Needed to release error strings.

    // StockLevel
    ICommandText*
    m_pIStockLevelCommand;
    HACCESSOR
    m_hStockLevelInputAccessor; // accessor
to bind input parameters
    HACCESSOR
    m_hStockLevelOutputAccessor; // accessor
to bind output columns
    DBPARAMS
    m_StockLevelExecuteParams; //
parameter structure for Execute

    // NewOrder
    // One prepared command for each
possible number of new order line items
    ICommandText*
    m_pINewOrderCommand[MAX_OL_NEW_ORDER_ITEMS]
;
// accessors to bind input
parameters
// one for each possible number
of new order line items
    HACCESSOR
    m_hNewOrderInputAccessor[MAX_OL_NEW_ORDER_I
TEMS];
// 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[MAX_OL_NEW_ORDER
_ITEMS];
// parameter structure for
Execute
    DBPARAMS
    m_NewOrderExecuteParams[MAX_OL_NEW_ORDER_IT
EMS];

    // 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);

```

trans.h

```

/* FILE: TRANS.H Microsoft
* TPC-C Kit Ver. 4.42.000 Copyright
* Microsoft, 2002 Copyrigh
* All Rights Reserved
* Version
4.10.000 audited by Richard Gimarc, Performance
Metrics, 3/17/99
* PURPOSE: Header file for TPC-C structure
templates.
* 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
*/
#pragma once

// 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 DATETIME_LEN 30
#define CREDIT_LEN 2
#define C_DATA_LEN 250
#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

// TIMESTAMP_STRUCT is provided by the ODBC header
file sqltypes.h, but is not available
// when compiling with dlib, so redefined here.
Note: we are using the symbol "_SQLTYPES"
// (declared in sqltypes.h) as a way to determine if
TIMESTAMP_STRUCT has been declared.
#ifndef _SQLTYPES
typedef struct
{
    /* SQLSMALLINT */ short
    /* unsigned short */ year;
    /* unsigned short */ month;
    /* unsigned short */ day;
    /* unsigned short */ hour;
    /* unsigned short */ minute;
    /* unsigned short */ second;
    /* unsigned long */ fraction;
} TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after
transaction completes
enum EXEC_STATUS
{
    eOK, // 0
    "Transaction committed."
    eInvalidItem, // 1 "Item number
is not valid."
    eDeliveryFailed // 2 "Delivery
Post Failed."
};

// transaction structures
typedef struct
{
    // input params
    long
ol_supply_w_id;
    long
ol_i_id;

```

```

short
ol_quantity;

// output params
char
ol_i_name[I_NAME_LEN+1];
char
ol_brand_generic[BRAND_LEN+1];
double
ol_i_price;
double
ol_amount;
short
ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
    // input params
    long                w_id;
    short               d_id;
    long                c_id;
    short               o_ol_cnt;

    // output params
    EXEC_STATUS
    exec_status_code;
    char
    c_last[LAST_NAME_LEN+1];
    char                c_credit[CREDIT_LEN+1];
    double              c_discount;
    double              w_tax;
    double              d_tax;
    long                o_id;
    short
    o_commit_flag;
    TIMESTAMP_STRUCT   o_entry_d;
    short              o_all_local;
    double              total_amount;
    OL_NEW_ORDER_DATA
    OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{
    // input params
    long
    w_id;
    short
    d_id;
    long
    c_id;
    short
    c_d_id;
    long
    c_w_id;
    double
    h_amount;
    char
    c_last[LAST_NAME_LEN+1];

    // output params
    EXEC_STATUS
    exec_status_code;

```

```

TIMESTAMP_STRUCT   h_date;
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];
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];
char
c_first[FIRST_NAME_LEN+1];
char
c_middle[MIDDLE_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];
TIMESTAMP_STRUCT   c_since;
char
c_credit[CREDIT_LEN+1];
double
c_credit_lim;
double
c_discount;
double
c_balance;
char
c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
    long
    ol_i_id;
    long
    ol_supply_w_id;
    short
    ol_quantity;
    double
    ol_amount;
    TIMESTAMP_STRUCT   ol_delivery_d;
} OL_ORDER_STATUS_DATA;

```

```

typedef struct
{
    // input params
    long                w_id;
    short               d_id;
    long                c_id;
    char
    c_last[LAST_NAME_LEN+1];

    // output params
    EXEC_STATUS
    exec_status_code;
    char
    c_first[FIRST_NAME_LEN+1];
    char
    c_middle[MIDDLE_NAME_LEN+1];
    double              c_balance;
    long                o_id;
    TIMESTAMP_STRUCT   o_entry_d;
    short               o_carrier_id;
    OL_ORDER_STATUS_DATA
    OL[MAX_OL_ORDER_STATUS_ITEMS];
    short               o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;

typedef struct
{
    // input params
    long                w_id;
    short               o_carrier_id;

    // output params
    EXEC_STATUS
    exec_status_code;
    SYSTEMTIME
    queue_time;
    long
    o_id[10];
    // id's of delivered
    orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery
transactions and for writing them to the delivery
server.
typedef struct _DELIVERY_TRANSACTION
{
    SYSTEMTIME
    queue;
    //time delivery transaction queued
    long
    w_id;
    //delivery warehouse
    short
    o_carrier_id;
    //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{
    // input params
    long
    w_id;
    short
    d_id;
    short
    o_id;
    short
    threshold;

    // output params

```

```

EXEC_STATUS
exec_status_code;
long
low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;

```

txn_base.h

```

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

#pragma once

// need to declare functions for import, unless
// define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
public:
    CTPCC_BASE(void) {};
    virtual ~CTPCC_BASE(void) {};

    virtual PNEW_ORDER_DATA
    BuffAddr_NewOrder() = 0;
    virtual PPAYMENT_DATA
    BuffAddr_Payment() = 0;
    virtual PDELIVERY_DATA
    BuffAddr_Delivery() = 0;
    virtual PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() = 0;
    virtual PORDER_STATUS_DATA
    BuffAddr_OrderStatus() = 0;

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

```

```

virtual void OrderStatus
() = 0;
};

```

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
LS
#define
_APS_NEXT_RESOURCE_VALU
E 202
#define
_APS_NEXT_COMMAND_VALU
E 32768
#define
_APS_NEXT_CONTROL_VALU
E 201
#define
_APS_NEXT_SYMED_VALU
E 106
#endif
#endif

```

resource_.h

```

//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1 101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 102
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101
#endif
#endif

```

Appendix B: Database Design

The TPC-C database was created with the following Transact-SQL scripts:

createdb.sql

```
-----
-- File:      CREATEDB.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2005
-----

SET ANSI_NULL_DFLT_OFF ON
GO

USE master
GO

-----
-- Create temporary table for timing
-----
IF EXISTS( SELECT name FROM sysobjects WHERE name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

CREATE TABLE tpcc_timer
    (start_date CHAR(30),
     end_date   CHAR(30))
GO

INSERT INTO tpcc_timer VALUES(0,0)
GO

-----
-- Store starting time
-----
UPDATE tpcc_timer
SET start_date = (SELECT CONVERT(Char(30), GETDATE(), 21))
GO

-----
-- create main database files
-----
CREATE DATABASE tpcc
```

```
ON PRIMARY
(
    NAME             = MSSQL_tpcc_root,
    FILENAME         = 'c:\MSSQL_tpcc_root.mdf',
    SIZE             = 8MB,
    FILEGROWTH       = 0),

FILEGROUP MSSQL_stk_fg
(
    NAME             = MSSQL_stk1,
    FILENAME         = 'c:\stk\stk1\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk2,
    FILENAME         = 'c:\stk\stk2\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk3,
    FILENAME         = 'c:\stk\stk3\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk4,
    FILENAME         = 'c:\stk\stk4\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk5,
    FILENAME         = 'c:\stk\stk5\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk6,
    FILENAME         = 'c:\stk\stk6\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk7,
    FILENAME         = 'c:\stk\stk7\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk8,
    FILENAME         = 'c:\stk\stk8\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk9,
    FILENAME         = 'c:\stk\stk9\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk10,
    FILENAME         = 'c:\stk\stk10\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk11,
    FILENAME         = 'c:\stk\stk11\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk12,
    FILENAME         = 'c:\stk\stk12\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
(
    NAME             = MSSQL_stk13,
    FILENAME         = 'c:\stk\stk13\'',
    SIZE             = 169550MB,
    FILEGROWTH       = 0),
```

```

FILEGROUP MSSQL_cust_fg
(
    NAME = MSSQL_cust1,
    FILENAME = 'c:\cust\cust1\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust2,
    FILENAME = 'c:\cust\cust2\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust3,
    FILENAME = 'c:\cust\cust3\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust4,
    FILENAME = 'c:\cust\cust4\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust5,
    FILENAME = 'c:\cust\cust5\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust6,
    FILENAME = 'c:\cust\cust6\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust7,
    FILENAME = 'c:\cust\cust7\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust8,
    FILENAME = 'c:\cust\cust8\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust9,
    FILENAME = 'c:\cust\cust9\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust10,
    FILENAME = 'c:\cust\cust10\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust11,
    FILENAME = 'c:\cust\cust11\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust12,
    FILENAME = 'c:\cust\cust12\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_cust13,
    FILENAME = 'c:\cust\cust13\' ,
    SIZE = 106550MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL_ol_fg
(
    NAME = MSSQL_ol1,
    FILENAME = 'c:\ol\ol1\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol2,

```

```

    FILENAME = 'c:\ol\ol2\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol3,
    FILENAME = 'c:\ol\ol3\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol4,
    FILENAME = 'c:\ol\ol4\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol5,
    FILENAME = 'c:\ol\ol5\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol6,
    FILENAME = 'c:\ol\ol6\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol7,
    FILENAME = 'c:\ol\ol7\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol8,
    FILENAME = 'c:\ol\ol8\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol9,
    FILENAME = 'c:\ol\ol9\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol10,
    FILENAME = 'c:\ol\ol10\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol11,
    FILENAME = 'c:\ol\ol11\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol12,
    FILENAME = 'c:\ol\ol12\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_ol13,
    FILENAME = 'c:\ol\ol13\' ,
    SIZE = 120550MB,
    FILEGROWTH = 0),
FILEGROUP MSSQL_misc_fg
(
    NAME = MSSQL_misc1,
    FILENAME = 'c:\misc\misc1\' ,
    SIZE = 29950MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc2,
    FILENAME = 'c:\misc\misc2\' ,
    SIZE = 29950MB,
    FILEGROWTH = 0),
(
    NAME = MSSQL_misc3,
    FILENAME = 'c:\misc\misc3\' ,
    SIZE = 29950MB,
    FILEGROWTH = 0),

```

```

(
    NAME                = MSSQL_misc4,
    FILENAME             = 'c:\misc\misc4\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc5,
    FILENAME             = 'c:\misc\misc5\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc6,
    FILENAME             = 'c:\misc\misc6\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc7,
    FILENAME             = 'c:\misc\misc7\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc8,
    FILENAME             = 'c:\misc\misc8\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc9,
    FILENAME             = 'c:\misc\misc9\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc10,
    FILENAME             = 'c:\misc\misc10\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc11,
    FILENAME             = 'c:\misc\misc11\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc12,
    FILENAME             = 'c:\misc\misc12\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_misc13,
    FILENAME             = 'c:\misc\misc13\',
    SIZE                 = 29950MB,
    FILEGROWTH           = 0)

LOG ON
(
    NAME                = MSSQL_tpcc_log_1,
    FILENAME             = 'E:',
    SIZE                 = 1999990MB,
    FILEGROWTH           = 0),
(
    NAME                = MSSQL_tpcc_log_2,
    FILENAME             = 'F:',
    SIZE                 = 153940MB,
    FILEGROWTH           = 0)

COLLATE Latin1_General_BIN
GO

-----
-- Store ending time
-----
UPDATE tpcc_timer
SET   end_date   = (SELECT CONVERT(CHAR(30), GETDATE(), 21))
GO

```

```

SELECT DATEDIFF(second,(SELECT start_date FROM tpcc_timer),(SELECT end_date FROM
tpcc_timer))
GO

-----
-- remove temporary table
-----
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

```

backupdev.sql

```

-----
--
-- File:      BACKUPDEV.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2005
--
-----

USE master
GO

-----
-- create backup devices
-----
EXEC sp_addumpdevice 'disk','tpccback11','N:\tpccback11.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback12','O:\tpccback12.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback13','P:\tpccback13.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback6','Q:\tpccback6.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback7','R:\tpccback7.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback8','S:\tpccback8.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback9','T:\tpccback9.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback10','U:\tpccback10.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback1','V:\tpccback1.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback2','W:\tpccback2.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback3','X:\tpccback3.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback4','Y:\tpccback4.dmp'
GO
EXEC sp_addumpdevice 'disk','tpccback5','Z:\tpccback5.dmp'
GO

```

backup.sql

```

-----
--
-----

```

```

-- File:    BACKUP.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.61
--          Copyright Microsoft, 2005
-----

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate, 21)

DUMP DATABASE tpcc TO tpccback1, tpccback2, tpccback3, tpccback4, tpccback5,
tpccback6, tpccback7, tpccback8, tpccback9, tpccback10, tpccback11, tpccback12,
tpccback13 WITH init, stats = 1

SELECT @enddate = GETDATE()
SELECT 'End date: ',
       CONVERT(VARCHAR(30),@enddate, 21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)

GO

```

restore.sql

```

-----
-- File:    RESTORE.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.61
--          Copyright Microsoft, 2005
-----

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate, 21)

LOAD DATABASE tpcc FROM tpccback1, tpccback2, tpccback3, tpccback4, tpccback5,
tpccback6, tpccback7, tpccback8, tpccback9, tpccback10, tpccback11, tpccback12,
tpccback13 WITH stats = 1, replace

SELECT @enddate = GETDATE()
SELECT 'End date: ',
       CONVERT(VARCHAR(30),@enddate, 21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)

GO

```

removedb.sql

```

-----
-- File:    REMOVEDB.SQL
-----

```

```

-----
--          Microsoft TPC-C Benchmark Kit Ver. 4.61
--          Copyright Microsoft, 2005
-----

USE master
GO

-----
-- remove any existing database and backup files
-----

EXEC sp_dbrremove tpcc, dropdev
GO

EXEC sp_dropdevice 'tpccback1'
GO
EXEC sp_dropdevice 'tpccback2'
GO
EXEC sp_dropdevice 'tpccback3'
GO
EXEC sp_dropdevice 'tpccback4'
GO
EXEC sp_dropdevice 'tpccback5'
GO
EXEC sp_dropdevice 'tpccback6'
GO
EXEC sp_dropdevice 'tpccback7'
GO
EXEC sp_dropdevice 'tpccback8'
GO
EXEC sp_dropdevice 'tpccback9'
GO
EXEC sp_dropdevice 'tpccback10'
GO
EXEC sp_dropdevice 'tpccback11'
GO
EXEC sp_dropdevice 'tpccback12'
GO
EXEC sp_dropdevice 'tpccback13'
GO

```

idxcuscl.sql

```

-----
-- File:    IDXCUSCL.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.68
--          Copyright Microsoft, 2006
-----
--          Creates clustered index on customer table
-----

USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

```



```

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'customer_c1' )
    DROP INDEX customer.customer_c1

CREATE UNIQUE CLUSTERED INDEX customer_c1 ON customer(c_w_id, c_d_id, c_id)
ON MSSQL_cust_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
    CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
    DATEDIFF(second, @startdate, @enddate)
GO

```

idxcusnc.sql

```

-----
-- File:      IDXCUSNC.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
--          Creates non-clustered index on customer table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
    CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'customer_nc1' )
    DROP INDEX customer.customer_nc1

CREATE UNIQUE NONCLUSTERED INDEX customer_nc1 ON customer(c_w_id, c_d_id, c_last,
c_first, c_id)
ON MSSQL_cust_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
    CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
    DATEDIFF(second, @startdate, @enddate)
GO

```

idxdiscl.sql

```

-----
-- File:      IDXDISCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
--          Creates clustered index on district table
-----
USE tpcc

```

```

GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
    CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'district_c1' )
    DROP INDEX district.district_c1

CREATE UNIQUE CLUSTERED INDEX district_c1 ON district(d_w_id, d_id)
WITH FILLFACTOR=100 ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
    CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
    DATEDIFF(second, @startdate, @enddate)
GO

```

idxitmcl.sql

```

-----
-- File:      IDXITMCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
--          Creates clustered index on item table
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
    CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'item_c1' )
    DROP INDEX item.item_c1

CREATE UNIQUE CLUSTERED INDEX item_c1 ON item(i_id)
ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
    CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
    DATEDIFF(second, @startdate, @enddate)
GO

```

idxhiscl.sql

```

-----
--
-- File:      IDXHISCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2006
--
--           Creates clustered index on history table
--
--           CAUTION: This index is only beneficial for systems
--           CAUTION: with 8 or more processors.
--           CAUTION: It may negatively impact performance on
--           CAUTION: systems with less than 8 processors.
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'history_cl' )
  DROP INDEX history.history_cl

CREATE UNIQUE CLUSTERED INDEX history_cl ON history(h_c_w_id, h_date, h_c_d_id,
h_c_id, h_amount)
  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxnodcl.sql

```

-----
--
-- File:      IDXNODCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2006
--
--           Creates clustered index on new-order table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'new_order_cl' )

```

```

DROP INDEX new_order.new_order_cl

CREATE UNIQUE CLUSTERED INDEX new_order_cl ON new_order(no_w_id, no_d_id, no_o_id)
  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxodlcl.sql

```

-----
--
-- File:      IDXODLCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2006
--
--           Creates clustered index on order-line table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'order_line_cl' )
  DROP INDEX order_line.order_line_cl

CREATE UNIQUE CLUSTERED INDEX order_line_cl ON order_line(ol_w_id, ol_d_id, ol_o_id,
ol_number)
  ON MSSQL_ol_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxordcl.sql

```

-----
--
-- File:      IDXORDCL.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2006
--
--           Creates clustered index on orders table
--
-----

```

```

-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'orders_cl' )
  DROP INDEX orders.orders_cl

CREATE UNIQUE CLUSTERED INDEX orders_cl ON orders(o_w_id, o_d_id, o_id)
  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxstkcl.sql

```

-----
-- File:   IDXSTKCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Creates clustered index on stock table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'stock_cl' )
  DROP INDEX stock.stock_cl

CREATE UNIQUE CLUSTERED INDEX stock_cl ON stock(s_i_id, s_w_id)
  ON MSSQL_stk_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxwarcl.sql

```

-----
-- File:   IDXWARCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Creates clustered index on warehouse table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'warehouse_cl' )
  DROP INDEX warehouse.warehouse_cl

CREATE UNIQUE CLUSTERED INDEX warehouse_cl ON warehouse(w_id)
  WITH FILLFACTOR=100 ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

tables.sql

```

-----
-- File:   TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- 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 MSSQL_misc_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 MSSQL_misc_fg
go

```

```

create table customer
(

```

```

    c_id          int,
    c_d_id        tinyint,

```

```

    c_w_id        int,
    c_discount    smallmoney,
    c_credit_lim  money,
    c_last        char(16),
    c_first       char(16),
    c_credit      char(2),
    c_balance     money,
    c_ytd_payment money,
    c_payment_cnt smallint,
    c_delivery_cnt smallint,
    c_street_1    char(20),
    c_street_2    char(20),
    c_city        char(20),
    c_state       char(2),
    c_zip         char(9),
    c_phone       char(16),
    c_since       datetime,
    c_middle      char(2),
    c_data        char(500)
) on MSSQL_cust_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 MSSQL_misc_fg
go

```

```

create table new_order
(

```

```

    no_o_id       int,
    no_d_id       tinyint,
    no_w_id       int

```

```

) on MSSQL_misc_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 MSSQL_misc_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 MSSQL_ol_fg
go

create table item
(
    i_id            int,
    i_name          char(24),
    i_price         smallmoney,
    i_data          char(50),
    i_im_id         int
) on MSSQL_misc_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 MSSQL_stk_fg
go

```

neword.sql

```

-----
--
-- File:      NEWORD.SQL
--           Microsoft TPC-C Benchmark Kit Ver. 4.68
--           Copyright Microsoft, 2006
--
--           Creates neworder stored procedure
--
--           Interface Level:      4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF

```

```

GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder' )
    DROP PROCEDURE tpcc_neworder
GO

CREATE PROCEDURE    tpcc_neworder
    @w_id            int,
    @d_id            tinyint,
    @c_id            int,
    @o_ol_cnt        tinyint,
    @o_all_local     tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

AS
DECLARE @w_tax          smallmoney,
        @d_tax          smallmoney,
        @c_last         char(16),
        @c_credit        char(2),
        @c_discount     smallmoney,
        @i_price         smallmoney,
        @i_name          char(24),
        @i_data          char(50),
        @o_entry_d       datetime,
        @remote_flag     int,
        @s_quantity       smallint,
        @s_data          char(50),
        @s_dist          char(24),
        @li_no           int,
        @o_id            int,
        @commit_flag     tinyint,
        @li_id           int,
        @li_s_w_id       int,
        @li_qty           smallint,
        @ol_number       int,
        @c_id_local      int

BEGIN

BEGIN TRANSACTION n

```

```

-----
-- get district tax and next 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 read)
    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,
        @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

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
-- lq 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, @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
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 != @ol_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 +

ELSE 1 END

CASE WHEN (info.w_id = @w_id) THEN 0

OUTPUT @o_id,
        @d_id,
        @w_id,
        info.lino,
        info.i_id,
        "dec 31, 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,@ol_qty1 UNION ALL
      SELECT @i_id2,@s_w_id2,2,@ol_qty2 UNION ALL
      SELECT @i_id3,@s_w_id3,3,@ol_qty3 UNION ALL
      SELECT @i_id4,@s_w_id4,4,@ol_qty4 UNION ALL

```



```

                SELECT @i_id5,@s_w_id5,5,@ol_qty5      UNION ALL
                SELECT @i_id6,@s_w_id6,6,@ol_qty6      UNION ALL
                SELECT @i_id7,@s_w_id7,7,@ol_qty7      UNION ALL
                SELECT @i_id8,@s_w_id8,8,@ol_qty8      UNION ALL
                SELECT @i_id9,@s_w_id9,9,@ol_qty9      UNION ALL
                SELECT @i_id10,@s_w_id10,10,@ol_qty10   UNION ALL
                SELECT @i_id11,@s_w_id11,11,@ol_qty11  UNION ALL
                SELECT @i_id12,@s_w_id12,12,@ol_qty12  UNION ALL
                SELECT @i_id13,@s_w_id13,13,@ol_qty13  UNION ALL
                SELECT @i_id14,@s_w_id14,14,@ol_qty14  UNION ALL
                SELECT @i_id15,@s_w_id15,15,@ol_qty15) AS
uol(iid,wid,lino,qty)
                ) AS ol(iid,wid,lino,ol_qty,rownum)
            INNER JOIN
            item (repeatableread) 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_ol_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_ol_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(repeatableread),
       customer(repeatableread)
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

-----
delivery.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

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

-----
nnull-txns.sql
-----
--
-- File:    NULL-TXNS.SQL
--          Microsoft TPC-C Benchmark Kit Ver. 4.68
--          Copyright Microsoft, 2006
--
-- This script will create stored procs which
-- accept the same parameters and return correctly
-- formed results sets to match the standard TPC-C
-- stored procs. Of course, the advantage is that
-- these stored procs place almost no load on
-- SQL Server and do not require a database.
--
--          Interface Level:    4.10.000
--
-----
USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_delivery' )
    DROP PROCEDURE tpcc_delivery
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder' )
    DROP PROCEDURE tpcc_neworder
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_orderstatus' )
    DROP PROCEDURE tpcc_orderstatus
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_payment' )
    DROP PROCEDURE tpcc_payment
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_stocklevel' )
    DROP PROCEDURE tpcc_stocklevel
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_version' )
    DROP PROCEDURE tpcc_version
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'order_line_null' )
    DROP PROCEDURE order_line_null
GO

CREATE PROCEDURE    tpcc_delivery

```

```

                @w_id          int,
                @o_carrier_id  smallint
AS
DECLARE @d_id          tinyint,
        @o_id          int,
        @c_id          int,
        @total         numeric(12,2),
        @oid1          int,
        @oid2          int,
        @oid3          int,
        @oid4          int,
        @oid5          int,
        @oid6          int,
        @oid7          int,
        @oid8          int,
        @oid9          int,
        @oid10         int,
        @delaytime     varchar(30)
-----
-- uniform random delay of 0 - 1 second; avg = 0.50
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND())*1.00) AS decimal(4,3)) AS
char(5))
WAITFOR delay @delaytime
SELECT 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001
GO
CREATE PROCEDURE tpcc_neworder
        @w_id          int,
        @d_id          tinyint,
        @c_id          int,
        @o_ol_cnt      tinyint,
        @o_all_local   tinyint,
        @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
        @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
        @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
        @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
        @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
        @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
        @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
        @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
        @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
        @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
        @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
        @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
        @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
        @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
        @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0
AS
DECLARE @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit        char(2),
        @c_discount     numeric(4,4),

```

```

        @i_price        numeric(5,2),
        @i_name         char(24),
        @o_entry_d      datetime,
        @li_no          int,
        @o_id           int,
        @commit_flag    tinyint,
        @li_id          int,
        @li_qty         smallint,
        @delaytime     varchar(30)
BEGIN
-----
-- uniform random delay of 0 - 0.6 second; avg = 0.3
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND())*0.60) AS decimal(4,3)) AS
char(5))
WAITFOR delay @delaytime
-----
-- process orderlines
-----
SELECT @commit_flag = 1,
        @li_no       = 0
WHILE (@li_no < @o_ol_cnt)
BEGIN
        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
        SELECT @li_no = @li_no + 1
        SELECT @i_price = 23.45, @li_qty = @li_no
        IF (@li_id = 999999)
        BEGIN
                SELECT ',,0,,',0,0
                SELECT @commit_flag = 0
        END
        ELSE
        BEGIN
                SELECT 'Item Name blah',
                        17,
                        'G',
                        @i_price,

```

```

        @i_price * @li_qty
    END
END

-----
-- return order data to client
-----
SELECT  @w_tax      = 0.1234,
        @d_tax      = 0.0987,
        @o_id       = 3001,
        @c_last     = 'BAROUGHTABLE',
        @c_discount = 0.2198,
        @c_credit   = 'GC',
        @o_entry_d  = GETDATE()

SELECT  @w_tax,
        @d_tax,
        @o_id,
        @c_last,
        @c_discount,
        @c_credit,
        @o_entry_d,
        @commit_flag

END
GO

CREATE PROCEDURE    tpcc_orderstatus
    @w_id          int,
    @d_id          tinyint,

    @c_id          int,
    @c_last        char(16) = ''

AS
DECLARE @c_balance    numeric(12,2),
        @c_first      char(16),
        @c_middle     char(2),
        @o_id         int,
        @o_entry_d    datetime,
        @o_carrier_id smallint,
        @ol_cnt       smallint,
        @delaytime    varchar(30)

-----
-- uniform random delay of 0 - 0.2 second; avg = 0.1
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS
char(5))

WAITFOR delay @delaytime

SELECT  @c_id        = 113,
        @c_balance   = -10.00,
        @c_first     = '8YCodgytqCj8',
        @c_middle    = 'OE',
        @c_last      = 'OUGHTOUGHTABLE',
        @o_id        = 3456,
        @o_entry_d   = GETDATE(),
        @o_carrier_id = 1

```

```

SELECT  @ol_cnt = (RAND() * 11) + 5

SET      ROWCOUNT @ol_cnt

SELECT  ol_supply_w_id,
        ol_i_id,
        ol_quantity,
        ol_amount,
        ol_delivery_d
FROM    order_line_null

SELECT  @c_id,
        @c_last,
        @c_first,
        @c_middle,
        @o_entry_d,
        @o_carrier_id,
        @c_balance,
        @o_id

GO

CREATE PROCEDURE    tpcc_payment
    @w_id          int,
    @c_w_id        int,
    @h_amount       numeric(6,2),
    @d_id          tinyint,
    @c_d_id        tinyint,
    @c_id          int,
    @c_last        char(16) = ''

AS
DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state    char(2),
        @w_zip      char(9),
        @w_name     char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city     char(20),
        @d_state    char(2),
        @d_zip      char(9),
        @d_name     char(10),
        @c_first    char(16),
        @c_middle   char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city     char(20),
        @c_state    char(2),
        @c_zip      char(9),
        @c_phone    char(16),
        @c_since    datetime,
        @c_credit   char(2),
        @c_credit_lim numeric(12,2),
        @c_balance  numeric(12,2),
        @c_discount numeric(4,4),
        @data       char(500),
        @c_data     char(500),
        @datetime   datetime,
        @w_ytd      numeric(12,2),
        @d_ytd      numeric(12,2),

```

```

@cnt          smallint,
@val          smallint,
@screen_data  char(200),
@d_id_local   tinyint,
@w_id_local   int,
@c_id_local   int,
@delaytime    varchar(30)

-----
-- uniform random delay of 0 - 0.3 second; avg = 0.15
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS
char(5))

WAITFOR delay @delaytime

SELECT @screen_data = ''

-----
-- get customer info and update balances
-----
SELECT @d_street_1 = 'rqSHHakqyV',
@d_street_2 = 'zZ98nW3BR2s',
@d_city = 'ArNr4GNFV9',
@d_state = 'aV',
@d_zip = '453511111'

-----
-- get warehouse data and update year-to-date
-----
SELECT @w_street_1 = 'rqSHHakqyV',
@w_street_2 = 'zZ98nW3BR2s',
@w_city = 'ArNr4GNFV9',
@w_state = 'aV',
@w_zip = '453511111'

SELECT @c_id = 123,
@c_balance = -10000.00,
@c_first = 'KmR03Xureb',
@c_middle = 'OE',
@c_last = 'BAROUGHTBAR',
@c_street_1 = 'QpGdOHjv8mR9vNI8V',
@c_street_2 = 'dzKoCObBqbc3yu',
@c_city = 'zAKZXdC037FQxq',
@c_state = 'QA',
@c_zip = '700311111',
@c_phone = '2967264064528555',
@c_credit = 'GC',
@c_credit_lim = 50000.00,
@c_discount = 0.3069,
@c_since = GETDATE(),
@datetime = GETDATE()

-----
-- return data to client
-----
SELECT @c_id,
@c_last,
@datetime,
@w_street_1,
@w_street_2,

```

```

@w_city,
@w_state,
@w_zip,
@d_street_1,
@d_street_2,
@d_city,
@d_state,
@d_zip,
@c_first,
@c_middle,
@c_street_1,
@c_street_2,
@c_city,
@c_state,
@c_zip,
@c_phone,
@c_since,
@c_credit,
@c_credit_lim,
@c_discount,
@c_balance,
@screen_data

GO

CREATE PROCEDURE tpcc_stocklevel
    @w_id int,
    @d_id tinyint,
    @threshold smallint
AS
DECLARE @delaytime varchar(30)

-----
-- uniform random delay of 0 - 3.6 second; avg = 1.8
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS
char(5))

WAITFOR delay @delaytime

SELECT 49
GO

CREATE PROCEDURE tpcc_version
AS
DECLARE @version char(8)

BEGIN
    SELECT @version = '4.10.000'
    SELECT @version AS 'Version'
END
GO

CREATE TABLE order_line_null (
    [ol_i_id] [int] NOT NULL ,
    [ol_supply_w_id] [int] NOT NULL ,
    [ol_delivery_d] [datetime] NOT NULL ,
    [ol_quantity] [smallint] NOT NULL ,
    [ol_amount] [numeric](6, 2) NOT NULL
) ON [PRIMARY]

```

GO

```
INSERT INTO order_line_null VALUES ( 101, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 102, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 103, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 104, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 105, 1, GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 106, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 107, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 108, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 109, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 110, 1, GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 111, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 112, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 113, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 114, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 115, 1, GETDATE(), 5, 123.45 )
GO
```

ordstat.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
          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

```

payment.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_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

```

stocklev.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(OORDER GROUP)
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

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('new_order') = id OR
object_id('item') = id
ORDER BY lockflags asc
GO

```

```

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc, 'auto update statistics', FALSE
EXEC sp_dboption tpcc, 'auto create statistics', FALSE
GO

DECLARE @db_id int,
@tbl_id int

SET @db_id = DB_ID('tpcc')
SET @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)

SET @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO

```

version.sql

```

-----
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Extracts current version of SQL Server
--
-----

```

```

USE master
GO

SELECT CONVERT(char(20), SERVERPROPERTY('ProductVersion')),
CONVERT(char(20), SERVERPROPERTY('ProductLevel')),
CONVERT(char(29), SERVERPROPERTY('Edition'))
GO

SELECT CONVERT(char(30), GETDATE(), 21)
GO

```

RunSQLCfg.sql

```

-----
-- File: RUNSQLCFG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Sets suggested runtime server configuration
--
-----

```

```

--          parameters          --
--          -----          --
EXEC sp_configure 'show advanced option', 1
GO

RECONFIGURE WITH OVERRIDE
GO

-----
-- change this value to approximately the number of connected users
-----
EXEC sp_configure 'max worker threads',255

-----
-- increase priority of user threads
-----
EXEC sp_configure 'priority boost',1

-----
-- disable automatic checkpointing
-----
EXEC sp_configure 'recovery interval',32767

-----
-- change to a mask appropriate for the number of processors on the server
-----
EXEC sp_configure 'affinity mask',0xf

-----
-- enable fibers
-----
EXEC sp_configure 'lightweight pooling',1
GO

RECONFIGURE WITH OVERRIDE
GO

```

VerifyTpccLoad.sql

```

-----
--          File:  VerifyTPCCLoad.SQL          --
--          Microsoft TPC-C Benchmark Kit Ver. 4.68          --
--          Copyright Microsoft, 2006          --
-----

SET NOCOUNT ON
PRINT ' '
SELECT CONVERT(Char(30), GETDATE(), 21)
PRINT ' '

USE tpcc
GO

IF EXISTS (SELECT name
           FROM sysobjects
           WHERE name = 'TPCC_INFO' AND
                 type = 'U')
DROP TABLE TPCC_INFO

```

```

GO
PRINT 'WAREHOUSE TABLE'
SELECT count_big(*)
FROM warehouse
GO

PRINT 'DISTRICT TABLE = (10 * No of warehouses)'
SELECT count_big(*)
FROM district
GO

PRINT 'ITEM TABLE = 100,000'
SELECT count_big(*)
FROM item
GO

PRINT 'CUSTOMER TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM customer
GO

PRINT 'ORDERS TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM orders
GO

PRINT 'HISTORY TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM history
GO

PRINT 'STOCK TABLE = (100,000 * No of warehouses)'
SELECT count_big(*)
FROM stock
GO

PRINT 'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'
SELECT count_big(*)
FROM order_line
GO

PRINT 'NEW_ORDER TABLE = (9000 * No of warehouses)'
SELECT count_big(*)
FROM new_order
GO

CREATE TABLE TPCC_INFO
(
  INFO_DATE          datetime,
  NUM_WAREHOUSE      bigint,
  WAREHOUSE_TARGET   bigint,
  NUM_DISTRICT       bigint,
  DISTRICT_TARGET    bigint,
  NUM_ITEM            bigint,
  ITEM_TARGET        bigint,
  NUM_CUSTOMER       bigint,
  CUSTOMER_TARGET    bigint,
  NUM_ORDERS         bigint,
  ORDERS_TARGET      bigint,
  ORDERS_TARGET_LOW  bigint,
  ORDERS_TARGET_HIGH bigint,
  NUM_ORDER_LINE     bigint,
)

```

```

ORDER_LINE_TARGET          bigint,
ORDER_LINE_TARGET_LOW     bigint,
ORDER_LINE_TARGET_HIGH    bigint,
NUM_NEW_ORDER             bigint,
NEW_ORDER_TARGET         bigint,
NEW_ORDER_TARGET_LOW     bigint,
NEW_ORDER_TARGET_HIGH    bigint,
NUM_HISTORY              bigint,
HISTORY_TARGET           bigint,
NUM_STOCK                bigint,
STOCK_TARGET             bigint)
GO

DECLARE @NUM_WAREHOUSE          bigint,
        @WAREHOUSE_TARGET      bigint,
        @NUM_DISTRICT          bigint,
        @DISTRICT_TARGET      bigint,
        @NUM_ITEM              bigint,
        @ITEM_TARGET           bigint,
        @NUM_CUSTOMER          bigint,
        @CUSTOMER_TARGET      bigint,
        @NUM_ORDERS            bigint,
        @ORDERS_TARGET         bigint,
        @ORDERS_TARGET_LOW    bigint,
        @ORDERS_TARGET_HIGH   bigint,
        @NUM_ORDER_LINE       bigint,
        @ORDER_LINE_TARGET     bigint,
        @ORDER_LINE_TARGET_LOW bigint,
        @ORDER_LINE_TARGET_HIGH bigint,
        @NUM_NEW_ORDER        bigint,
        @NEW_ORDER_TARGET     bigint,
        @NEW_ORDER_TARGET_LOW  bigint,
        @NEW_ORDER_TARGET_HIGH bigint,
        @NUM_HISTORY           bigint,
        @HISTORY_TARGET        bigint,
        @NUM_STOCK             bigint,
        @STOCK_TARGET          bigint

-- set the local variables prior to inserting them into the TPCC_INFO table
SELECT @NUM_WAREHOUSE = COUNT_BIG(*)
FROM   warehouse

SELECT @NUM_DISTRICT = COUNT_BIG(*)
FROM   district

SELECT @NUM_ITEM = COUNT_BIG(*)
FROM   item

SELECT @NUM_CUSTOMER = COUNT_BIG(*)
FROM   customer

SELECT @NUM_ORDERS = COUNT_BIG(*)
FROM   orders

SELECT @NUM_ORDER_LINE = COUNT_BIG(*)
FROM   order_line

SELECT @NUM_NEW_ORDER = COUNT_BIG(*)
FROM   new_order

SELECT @NUM_HISTORY = COUNT_BIG(*)

```

```

FROM   history

SELECT @NUM_STOCK = COUNT_BIG(*)
FROM   stock

--- now calculate and set the target values
SELECT @WAREHOUSE_TARGET = @NUM_WAREHOUSE,
       @DISTRICT_TARGET  = @NUM_WAREHOUSE * 10,
       @ITEM_TARGET      = 100000,
       @CUSTOMER_TARGET  = @NUM_WAREHOUSE * 30000,
       @ORDERS_TARGET    = @NUM_WAREHOUSE * 30000,
       @ORDERS_TARGET_LOW = @ORDERS_TARGET - FLOOR(@ORDERS_TARGET * .01),
       @ORDERS_TARGET_HIGH = @ORDERS_TARGET + FLOOR(@ORDERS_TARGET * .01),
       @ORDER_LINE_TARGET = @NUM_WAREHOUSE * 300000,
       @ORDER_LINE_TARGET_LOW = @ORDER_LINE_TARGET - FLOOR(@ORDER_LINE_TARGET *
.01),
       @ORDER_LINE_TARGET_HIGH = @ORDER_LINE_TARGET + FLOOR(@ORDER_LINE_TARGET *
.01),
       @NEW_ORDER_TARGET = @NUM_WAREHOUSE * 9000,
       @NEW_ORDER_TARGET_LOW = @NEW_ORDER_TARGET - FLOOR(@NEW_ORDER_TARGET *
.01),
       @NEW_ORDER_TARGET_HIGH = @NEW_ORDER_TARGET + FLOOR(@NEW_ORDER_TARGET *
.01),
       @HISTORY_TARGET    = @NUM_WAREHOUSE * 30000,
       @STOCK_TARGET      = @NUM_WAREHOUSE * 100000

--- insert the values into TPCC_INFO
INSERT INTO TPCC_INFO VALUES (GETDATE(),
                              @NUM_WAREHOUSE,
                              @WAREHOUSE_TARGET,
                              @NUM_DISTRICT,
                              @DISTRICT_TARGET,
                              @NUM_ITEM,
                              @ITEM_TARGET,
                              @NUM_CUSTOMER,
                              @CUSTOMER_TARGET,
                              @NUM_ORDERS,
                              @ORDERS_TARGET,
                              @ORDERS_TARGET_LOW,
                              @ORDERS_TARGET_HIGH,
                              @NUM_ORDER_LINE,
                              @ORDER_LINE_TARGET,
                              @ORDER_LINE_TARGET_LOW,
                              @ORDER_LINE_TARGET_HIGH,
                              @NUM_NEW_ORDER,
                              @NEW_ORDER_TARGET,
                              @NEW_ORDER_TARGET_LOW,
                              @NEW_ORDER_TARGET_HIGH,
                              @NUM_HISTORY,
                              @HISTORY_TARGET,
                              @NUM_STOCK,
                              @STOCK_TARGET)

GO

--- output the row counts from the build
PRINT ''
PRINT ''
PRINT '-----'
PRINT '|   WAREHOUSE TABLE   |'
PRINT '-----'
SELECT TOP 1

```

```

CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_WAREHOUSE AS 'Warehouse Rows',
WAREHOUSE_TARGET AS 'Warehouse Target',
CASE WHEN (NUM_WAREHOUSE = WAREHOUSE_TARGET)
THEN 'OK!'
ELSE 'ERROR!!!'
END AS 'Warehouse Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| DISTRICT TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_DISTRICT AS 'District Rows',
DISTRICT_TARGET AS 'District Target',
CASE WHEN (NUM_DISTRICT = DISTRICT_TARGET)
THEN 'OK!'
ELSE 'ERROR!!!'
END AS 'District Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| ITEM TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_ITEM AS 'Item Rows',
ITEM_TARGET AS 'Item Target',
CASE WHEN (NUM_ITEM = ITEM_TARGET)
THEN 'OK!'
ELSE 'ERROR!!!'
END AS 'Item Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| CUSTOMER TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_CUSTOMER AS 'Customer Rows',
CUSTOMER_TARGET AS 'Customer Target',
CASE WHEN (NUM_CUSTOMER = CUSTOMER_TARGET)
THEN 'OK!'
ELSE 'ERROR!!!'
END AS 'Customer Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'

```

```

PRINT '| ORDERS TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_ORDERS AS 'Orders Rows',
ORDERS_TARGET AS 'Orders Target',
CASE WHEN (NUM_ORDERS = ORDERS_TARGET)
THEN 'OK!'
WHEN (NUM_ORDERS BETWEEN ORDERS_TARGET_LOW AND ORDERS_TARGET_HIGH)
THEN 'OK! (within 1%)'
ELSE 'ERROR!!!'
END AS 'Orders Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| ORDER LINE TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_ORDER_LINE AS 'Order Line Rows',
ORDER_LINE_TARGET AS 'Order Line Target',
CASE WHEN (NUM_ORDER_LINE = ORDER_LINE_TARGET)
THEN 'OK!'
WHEN (NUM_ORDER_LINE BETWEEN ORDER_LINE_TARGET_LOW AND
ORDER_LINE_TARGET_HIGH)
THEN 'OK! (within 1%)'
ELSE 'ERROR!!!'
END AS 'Order Line Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| NEW ORDER TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_NEW_ORDER AS 'New Order Rows',
NEW_ORDER_TARGET AS 'New Order Target',
CASE WHEN (NUM_NEW_ORDER = NEW_ORDER_TARGET)
THEN 'OK!'
WHEN (NUM_NEW_ORDER BETWEEN NEW_ORDER_TARGET_LOW AND
NEW_ORDER_TARGET_HIGH)
THEN 'OK! (within 1%)'
ELSE 'ERROR!!!'
END AS 'New Order Message'
FROM TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '| HISTORY TABLE |'
PRINT '-----'
SELECT TOP 1
CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
NUM_HISTORY AS 'History Rows',

```

```

        HISTORY_TARGET          AS      'History Target',
        CASE WHEN (NUM_HISTORY = HISTORY_TARGET)
            THEN 'OK!'
            ELSE 'ERROR!!!'
        END                      AS      'History Message'
FROM    TPCC_INFO
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '|          STOCK TABLE          |'
PRINT '-----'
SELECT TOP 1
        CONVERT(Char(30),INFO_DATE,21) AS 'Date',
        NUM_STOCK                      AS 'Stock Rows',
        STOCK_TARGET                   AS 'Stock Target',
        CASE WHEN (NUM_STOCK = STOCK_TARGET)
            THEN 'OK!'
            ELSE 'ERROR!!!'
        END                      AS      'Stock Message'
FROM    TPCC_INFO
GO

-----
-- Check Indexes
-----
USE tpcc
GO

PRINT ''
PRINT ''
PRINT '-----'
PRINT '|          TPC-C INDEXES          |'
PRINT '-----'
EXEC sp_helpindex warehouse
EXEC sp_helpindex district
EXEC sp_helpindex item
EXEC sp_helpindex customer
EXEC sp_helpindex orders
EXEC sp_helpindex order_line
EXEC sp_helpindex new_order
EXEC sp_helpindex history
EXEC sp_helpindex stock
GO

```

sqlshutdown.sql

```

-----
-- File:      SQLSHUTDOWN.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.68
--            Copyright Microsoft, 2006
--
--            Checkpoints tpcc database and issues a shutdown
--
-----
USE tpcc
GO

```

```

CHECKPOINT
GO

SHUTDOWN
GO

```

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, TPCC_LDR_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  = INDEX_ORDER;
    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();
exit(1);
}

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)
pargs->table_customer =
TRUE;
else if (strcmp(ptr+2,"orders") ==
0)
pargs->table_orders =
TRUE;
}
}

```

```

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

break;
}

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

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

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

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

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

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

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

default:
GetArgsLoaderUsage();
exit(-1);
break;
}

}

/* check for required args */
if (pargs->num_warehouses == UNDEF )
{
printf("Number of Warehouses is required\n");
exit(-2);
}

return;
}

//=====
//
// Function name: GetArgsLoaderUsage
//
//=====

```

```

void GetArgsLoaderUsage()
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId());
#endif

    printf("TPCC_LDR:\n\n");
    printf("Parameter                                     Default\n");
    printf("-----\n");
    printf("-W Number of Warehouses to Load                    Required \n");
    printf("-S Server                                             %s\n", SERVER);
    printf("-U Username                                           %s\n", USER);
    printf("-P Password                                           %s\n", PASSWORD);
    printf("-D Database                                           %s\n", DATABASE);
    printf("-b Batch Size                                         %ld\n",
(long) BATCH);
    printf("-p TDS packet size                                   %ld\n",
(long) DEF_LD_PACKET_SIZE);
    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                               %ld\n",
(long) DEF_STARTING_WAREHOUSE);
    printf("-i Build Option (data = 0, data and index = 1)     %ld\n",
(long) BUILD_INDEX);
    printf("-o Cluster Index Build Order (before = 1, after = 0) %ld\n",
(long) INDEX_ORDER);
    printf("-c Build Scaled Database (normal = 0, tiny = 1)   %ld\n",
(long) SCALE_DOWN);
    printf("-d Index Script Path                                 %s\n",
INDEX_SCRIPT_PATH);
    printf("-t Table to Load                                     all tables\n");
    printf(" [item|warehouse|customer|orders]\n");
    printf(" Notes: \n");
    printf(" - the '-t' parameter may be included multiple times to \n");
    printf(" specify multiple tables to be loaded \n");
    printf(" - 'item' loads ITEM table \n");
    printf(" - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf(" - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf(" - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

    printf("\nNote: Command line switches are case sensitive.\n");

    exit(0);
}

```

random.c

```

// File: RANDOM.C
// Microsoft TPC-C Kit Ver. 4.62
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000, 2001, 2002, 2005

```

```

// Purpose: Random number generation routines for database loader

// Includes
#include "tpcc.h"
#include "math.h"

// Defines
#define A 16807
#define M 2147483647
#define Q 127773 /* M div A */
#define R 2836 /* M mod A */
#define Thread __declspec(thread)

// Globals
long Thread Seed = 0; /* thread local seed */

/*****
 * random -
 * Implements a GOOD pseudo random number generator. This generator
 * will/should? run the complete period before repeating.
 *
 * Copied from:
 * Random Numbers Generators: Good Ones Are Hard to Find.
 * Communications of the ACM - October 1988 Volume 31 Number 10
 *
 * Machine Dependencies:
 * long must be 2 ^ 31 - 1 or greater.
 *
 *****/

/*****
 * seed - load the Seed value used in irand and drand. Should be used before
 * first call to irand or drand.
 *****/

void seed(long val)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering seed()...\n", (int) GetCurrentThreadId());
    printf("Old Seed %ld New Seed %ld\n", Seed, val);
#endif

    if ( val < 0 )
        val = abs(val);

    Seed = val;
}

/*****
 * irand - returns a 32 bit integer pseudo random number with a period of
 * 1 to 2 ^ 32 - 1.
 *
 * parameters:
 * none.
 *****/

```



```

* returns:                                     *
*   32 bit integer - defined as long ( see above ). *
*
* side effects:                                 *
*   seed get recomputed.                       *
*****/
long irand()
{
    register long    s;      /* copy of seed */
    register long    test;   /* test flag */
    register long    hi;     /* tmp value for speed */
    register long    lo;     /* tmp value for speed */

#ifdef DEBUG
    printf("[%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 %ld & %ld ==> %ld\n",
           (int) GetCurrentThreadId(), lower, upper,
           rand_num);
#endif

    return rand_num;
}
#endif

//=====
// Function   : NURand
//
// Description:
//=====

```

```

long NURand(int iConst,
           long x,
           long y,
           long C)
{
    long rand_num;

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

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

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(), rand_num);
#endif

    return rand_num;
}

```

strings.c

```

// File: STRINGS.C
// Microsoft TPC-C Kit Ver. 4.51
// Copyright Microsoft, 1996, 1997, 1998, 1999,
// 2000, 2001, 2002, 2003
// Purpose: Source file for database loader string functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
// Function name: MakeAddress
//=====

void MakeAddress(char *street_1,
                char *street_2,
                char *city,
                char *state,
                char *zip)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering MakeAddress()\n", (int) GetCurrentThreadId());
#endif

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG

```

```

    printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state: %s,
zip: %s\n",
           (int) GetCurrentThreadId(), street_1, street_2, city,
state, zip);
#endif

    return;
}

//=====
// Function name: LastName
//=====

void LastName(int num,
             char *name)
{
    static char *n[] =
    {
        "BAR" , "OUGHT" , "ABLE" , "PRI" , "PRES",
        "ESE" , "ANTI" , "CALLY" , "ATION", "EING"
    };

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

    if ((num >= 0) && (num < 1000))
    {
        strcpy(name, n[(num/100)%10]);
        strcat(name, n[(num/10)%10]);
        strcat(name, n[(num/1)%10]);

        if (strlen(name) < LAST_NAME_LEN)
        {
            PaddString(LAST_NAME_LEN, name);
        }
    }
    else
    {
        printf("\nError in LastName()... num <%ld> out of range
(0,999)\n", num);
        exit(-1);
    }

#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
           (int) GetCurrentThreadId(), num, num/100, (num/10)%10,
num%10);
    printf("[%ld]DBG: LastName: String = %s\n", (int) GetCurrentThreadId(),
name);
#endif

    return;
}

```

```

//=====
//
// Function name: MakeAlphaString
//
//=====

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9. The only other
//requirement is that the character set used "must be able to represent a minimum
//of 128 different characters". We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//--CLevine 08/13/96

int MakeAlphaString( int x, int y, int z, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

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

    len= RandomNumber(x, y);

    for (i=0; i<len; i++)
        str[i] = chArray[RandomNumber(0,chArrayMax)];
    str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int minLen, int maxLen, int padLen, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

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

    len= RandomNumber(minLen, maxLen);

    for (i=0; i<len; i++)
        str[i] = chArray[RandomNumber(0,chArrayMax)];
    if (len < padLen)

```

```

        memset(str+len, ' ', padLen - len);
        str[padLen] = 0;
    }
    return padLen;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====

int MakeOriginalAlphaString(int x,
                            int y,
                            int z,
                            char *str,
                            int percent)
{
    int len;
    int val;
    int start;

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

    // verify 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);
    }

    // Make Alpha String
    len = 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("[%ld]DBG: MakeOriginalAlphaString: : %s\n",
(int) GetCurrentThreadId(), str);
#endif

    return len;
}

```

```

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

    //MakeNumberString is always called MakeZipNumberString(16, 16, 16,
string)

    memset(str, '0', 16);
    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str+8, tmp, strlen(tmp));

    str[16] = 0;

    return 16;
}

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

    //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9,
string)

    strcpy(str, "000011111");

    itoa(RandomNumber(0, 9999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    return 9;
}

//=====
//
// Function name: InitString
//
//=====
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

    memset(str, ' ', len);
    str[len] = 0;
}

```

```

}

//=====
// Function name: InitAddress
//
// Description:
//
//=====
void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
    memset(street_1, ' ', ADDRESS_LEN+1);
    memset(street_2, ' ', ADDRESS_LEN+1);
    memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1] = 0;

    memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;

    memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}

//=====
//
// Function name: PaddString
//
//=====
void PaddString(int max, char *name)
{
    int len;

    len = strlen(name);
    if ( len < max )
        memset(name+len, ' ', max - len);
    name[max] = 0;

    return;
}


```

time.c

```

// File: TIME.C
// Microsoft TPC-C Kit Ver. 4.62
// Copyright Microsoft, 1996, 1997, 1998, 1999,
2000, 2001, 2002, 2005
// Purpose: Source file for time functions

// Includes
#include "tpcc.h"

// Globals
static long start_sec;

```

```

//=====
//
// Function name: TimeNow
//
//=====
long TimeNow()
{
    long            time_now;
    struct _timeb  el_time;

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

    _ftime(&el_time);

    time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

    return time_now;
}

```

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       "C:\\MSTPCC.450\\SETUP\\LOGS\\load.out"
#define LOADER_LOG_PATH       "C:\\MSTPCC.450\\SETUP\\LOGS\\"
#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;
    BOOL                    // set if loading all tables
                                table_item;
    BOOL                    // set if loading ITEM table specifically
                                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 District();
void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();
void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void CheckForCommit_Big();
void OpenConnections();
void BuildIndex();
void FormatDate ();

// Shared memory structures
typedef struct
{
    double ol_i_id; ol;
    long ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
}

```

```

    char          ol_dist_info[DIST_INFO_LEN+1];
    char
    ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long          o_id;
    short        o_d_id;
    long         o_w_id;
    long         o_c_id;
    short        o_carrier_id;
    short        o_ol_cnt;
    short        o_all_local;
    ORDER_LINE_STRUCT  o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long          c_id;
    short        c_d_id;
    long         c_w_id;
    char         c_first[FIRST_NAME_LEN+1];
    char         c_middle[MIDDLE_NAME_LEN+1];
    char         c_last[LAST_NAME_LEN+1];
    char         c_street_1[ADDRESS_LEN+1];
    char         c_street_2[ADDRESS_LEN+1];
    char         c_city[ADDRESS_LEN+1];
    char         c_state[STATE_LEN+1];
    char         c_zip[ZIP_LEN+1];
    char         c_phone[PHONE_LEN+1];
    char         c_credit[CREDIT_LEN+1];
    double       c_credit_lim;
    double       c_discount;
    char         c_balance[6];
    double       c_ytd_payment;
    short        c_payment_cnt;
    short        c_delivery_cnt;
    char         c_data[C_DATA_LEN+1];
    double       h_amount;
    char         h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
    char         c_last[LAST_NAME_LEN+1];
    char         c_first[FIRST_NAME_LEN+1];
    long         c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long         time_start;
} LOADER_TIME_STRUCT;

// Global variables
char          szLastError[300];
HENV         henv;

```

```

HDBC          v_hdbc; // for SQL
Server version verification
HDBC          i_hdbc1; // for ITEM table
HDBC          w_hdbc1; // for WAREHOUSE,
DISTRICT, STOCK
HDBC          c_hdbc1; // for CUSTOMER
HDBC          c_hdbc2; // for HISTORY
HDBC          o_hdbc1; // for ORDERS
HDBC          o_hdbc2; // for NEW-ORDER

HDBC          o_hdbc3; // for ORDER-LINE

HSTMT         v_hstmt; // for SQL Server
version verification
HSTMT         i_hstmt1;
HSTMT         w_hstmt1;
HSTMT         c_hstmt1, c_hstmt2;
HSTMT         o_hstmt1, o_hstmt2, o_hstmt3;

int           total_db_errors;

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;

TPCC_LDR_ARGS *aptr, args;

//=====
//
// Function name: main
//
//=====
int main(int argc, char **argv)
{
    DWORD       dwThreadID[MAX_MAIN_THREADS];
    HANDLE      hThread[MAX_MAIN_THREADS];
    FILE        *fLoader;
    char        buffer[255];
    int         i;

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

    printf("\n*****");
    printf("\n*");
}

```

```

printf("\n* Microsoft SQL Server          *");
printf("\n*                               *");
printf("\n* TPC-C BENCHMARK KIT: Database loader *");
printf("\n* Version %s                          *", TPCKIT_VER);
printf("\n*                               *");
printf("\n*****\n\n");

// process command line arguments
aptr = &args;
GetArgsLoader(argc, argv, aptr);

printf("Build interface is ODBC.\n");

if (aptr->build_index == 0)
    printf("Data load only - no index creation.\n");
else
    printf("Data load and index creation.\n");

if (aptr->index_order == 0)
    printf("Clustered indexes will be created after bulk load.\n");
else
    printf("Clustered indexes will be created before bulk load.\n");

// 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.\n", aptr->num_warehouses);
if (aptr->scale_down == 1)
{
    sprintf(buffer, "SCALED DOWN DATABASE.\n");
}

```

```

printf("%s", buffer);
fprintf(fLoader, "%s", buffer);

main_time_start = (TimeNow() / MILLI);

// start parallel load threads
if (aptr->tables_all || aptr->table_item)
{
    fprintf(fLoader, "\nStarting loader threads for: item\n");
    hThread[0] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadItem,
                                NULL,
                                0,
                                &dwThreadID[0]);
    if (hThread[0] == NULL)
    {
        printf("Error, failed in creating creating thread =
0.\n");
        exit(-1);
    }
    if (aptr->tables_all || aptr->table_warehouse)
    {
        fprintf(fLoader, "Starting loader threads for: warehouse\n");
        hThread[1] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadWarehouse,
                                NULL,
                                0,
                                &dwThreadID[1]);
        if (hThread[1] == NULL)
        {
            printf("Error, failed in creating creating thread =
1.\n");
            exit(-1);
        }
    }
    if (aptr->tables_all || aptr->table_customer)
    {
        fprintf(fLoader, "Starting loader threads for: customer\n");
        hThread[2] = CreateThread(NULL,
                                0,
                                (LPTHREAD_START_ROUTINE) LoadCustomer,
                                NULL,

```



```

0,
&dwThreadID[2]);
    if (hThread[2] == NULL)
    {
        printf("Error, failed in creating creating main thread
= 2.\n");
        exit(-1);
    }
    if (aptr->tables_all || aptr->table_orders)
    {
        fprintf(fLoader, "Starting loader threads for: orders\n");
        hThread[3] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadOrders,
NULL,
0,
&dwThreadID[3]);
        if (hThread[3] == NULL)
        {
            printf("Error, failed in creating creating main thread
= 3.\n");
            exit(-1);
        }
        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
            if (hThread[i] != NULL)
            {
                WaitForSingleObject( hThread[i], INFINITE );
                CloseHandle(hThread[i]);
                hThread[i] = NULL;
            }
        }
        main_time_end = (TimeNow() / MILLI);
        sprintf(buffer, "\nTPC-C load completed successfully in %ld minutes.\n",
            (main_time_end - main_time_start)/60);
        printf("%s",buffer);
        fprintf(fLoader, "%s", buffer);
        fclose(fLoader);
        SQLFreeEnv(henv);
        exit(0);
        return 0;
    }

```

```

//=====
//
// Function name: LoadItem
//
//=====
void LoadItem()
{
    int          i;
    long         i_id;
    long         i_im_id;
    char         i_name[I_NAME_LEN+1];
    double       i_price;
    char         i_data[I_DATA_LEN+1];
    char         name[20];
    long         time_start;
    RETCODE      rc;
    DBINT        rcint;
    char         bcphint[128];
    char         err_log_path[256];

    // Seed with unique number
    seed(11);

    printf("Loading item table...\n");

    //if build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxitm1");

    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(bcphint, "tablock, order (i_id), ROWS_PER_BATCH =
100000");
        rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
        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 != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, SQL_VARLEN_DATA, "", 1, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(i_hdbc1);

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

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

            MakeAlphaStringPadded(14, 24, I_NAME_LEN, i_name);

            i_price = ((float) RandomNumber(100L, 10000L))/100.0;

            MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

            rc = bcp_sendrow(i_hdbc1);
            if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

            item_rows_loaded++;
            CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item",
&time_start);
        }

        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
            HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
            BuildIndex("idxitm1");
    }

//=====
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//=====
void LoadWarehouse()
{
    int         i;
    long        w_id;

```

```

char    w_name[W_NAME_LEN+1];
char    w_street_1[ADDRESS_LEN+1];
char    w_street_2[ADDRESS_LEN+1];
char    w_city[ADDRESS_LEN+1];
char    w_state[STATE_LEN+1];
char    w_zip[ZIP_LEN+1];
double  w_tax;
double  w_ytd;
char    name[20];
long    time_start;
RETCODE rc;
DBINT   rcint;
char    bcphint[128];
char    err_log_path[256];

// Seed with unique number
seed(2);

printf("Loading warehouse table...\n");

// if build index before load...
if ((aptr->build_index == 1) && (aptr->index_order == 1))
    BuildIndex("idxwarcl");

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");
rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);

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

if ((aptr->build_index == 1) && (aptr->index_order == 1))
{
    sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d",
aptr->num_warehouses);
    rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
}

i = 0;
rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

```

```

++i); rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
++i); rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0, STATE_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);
rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0, ZIP_LEN, NULL, 0, 0, ++i);
if (rc != SUCCEED)
    HandleErrorDBC(w_hdbc1);

time_start = (TimeNow() / MILLI);
warehouse_rows_loaded = 0;

for (w_id = (long)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
{
    MakeAlphaStringPadded(6,10, W_NAME_LEN, w_name);
    MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);
    w_tax = ((float) RandomNumber(0L,2000L))/10000.00;
    w_ytd = 300000.00;

    rc = bcp_sendrow(w_hdbc1);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    warehouse_rows_loaded++;
    CheckForCommit(w_hdbc1, i_hstmt1, warehouse_rows_loaded,
"warehouse", &time_start);
}

rcint = bcp_done(w_hdbc1);
if (rcint < 0)
    HandleErrorDBC(w_hdbc1);

printf("Finished loading warehouse table.\n");

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxwarc1");

stock_rows_loaded = 0;
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 bcp_hint[128];
    char err_log_path[256];

    // Seed with unique number
    seed(4);

    printf("Loading district table...\n");

    // build index before load
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxdisc1");

    InitString(d_name, D_NAME_LEN+1);
    InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
    sprintf(name, "%s..%s", aptr->database, "district");

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "district.err");
    rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcp_hint, "tablock, order (d_w_id, d_id), ROWS_PER_BATCH
= %u", (aptr->num_warehouses * 10));
        rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcp_hint);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(w_hdbc1);
SQLFLT8, ++i);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
SQLINT4, ++i);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
SQLFLT8, ++i);
        rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0, SQL_VARLEN_DATA, NULL, 0,
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0, D_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0, ADDRESS_LEN, NULL, 0, 0,
        ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0, ADDRESS_LEN, NULL, 0, 0,
        ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        d_ytd = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses;
        w_id++)
        {
            d_w_id = w_id;

            for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
            {
                MakeAlphaStringPadded(6,10,D_NAME_LEN, d_name);

                MakeAddress(d_street_1, d_street_2, d_city, d_state,
                d_zip);

                d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

                district_rows_loaded++;
                CheckForCommit(w_hdbc1, w_hstmt1,
                district_rows_loaded, "district", &time_start);

```

```

    }

    rcint = bcp_done(w_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(w_hdbc1);

    printf("Finished loading district table.\n");

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxdiscl");

    return;
}

//=====
//
// Function : Stock
//
//=====
void Stock()
{
    int i;
    long s_i_id;
    long s_w_id;
    short s_quantity;
    char s_dist_01[S_DIST_LEN+1];
    char s_dist_02[S_DIST_LEN+1];
    char s_dist_03[S_DIST_LEN+1];
    char s_dist_04[S_DIST_LEN+1];
    char s_dist_05[S_DIST_LEN+1];
    char s_dist_06[S_DIST_LEN+1];
    char s_dist_07[S_DIST_LEN+1];
    char s_dist_08[S_DIST_LEN+1];
    char s_dist_09[S_DIST_LEN+1];
    char s_dist_10[S_DIST_LEN+1];
    long s_ytd;
    short s_order_cnt;
    short s_remote_cnt;
    char s_data[S_DATA_LEN+1];
    short len;
    char name[20];
    long time_start;
    RETCODE rc;
    DBINT rcint;
    char bcp_hint[128];
    char err_log_path[256];

    // Seed with unique number
    seed(3);

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
        BuildIndex("idxstkc1");

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

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "stock.err");
    rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);

```

```

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

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
            sprintf(bcphint, "tablock, order (s_i_id, s_w_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 10000));
            rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
            if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT2, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, SQL_VARLEN_DATA, "", 1, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

```

```

        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);

        s_ytd = s_order_cnt = s_remote_cnt = 0;

        time_start = (TimeNow() / MILLI);

        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 <= aptr->
num_warehouses; s_w_id++)
            {
                s_quantity = (short)RandomNumber(10L,100L);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                len = MakeOriginalAlphaString(26,50, S_DATA_LEN,
s_data,10);

                rc = bcp_sendrow(w_hdbc1);
                if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

                stock_rows_loaded++;
                CheckForCommit_Big(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);
            }
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)

```

```

        HandleErrorDBC(w_hdbc1);

printf("Finished loading stock table.\n");

SQLFreeStmt(w_hstmt1, SQL_DROP);
SQLDisconnect(w_hdbc1);
SQLFreeConnect(w_hdbc1);

// if build index after load...
if ((aptr->build_index == 1) && (aptr->index_order == 0))
    BuildIndex("idxstkcl");

return;
}

//=====
//
// Function   : LoadCustomer
//
//=====
void LoadCustomer()
{
    LOADER_TIME_STRUCT    customer_time_start;
    LOADER_TIME_STRUCT    history_time_start;
    long                  w_id;
    short                 d_id;
    DWORD                  dwThreadID[MAX_CUSTOMER_THREADS];
    HANDLE                  hThread[MAX_CUSTOMER_THREADS];
    char                   name[20];
    RETCODE                rc;
    DBINT                  rcint;
    char                   bcphint[128];
    char                   cmd[256];
    int                     num_procs;
    char                   err_log_path_cust[256];
    char                   err_log_path_hist[256];

    // Seed with unique number
    seed(5);

    printf("Loading customer and history tables...\n");

    // if build index before load...
    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        BuildIndex("idxcuscl");
        // check the number of processors on this system
        // if 8 or more processors, then build index on History.
        // if less than 8 processors, do not build the index
        num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
        if ( num_procs >= 8 )
            BuildIndex("idxhiscl");
    }

    // Initialize bulk copy
    sprintf(name, "%s..%s", aptr->database, "customer");

    strcpy(err_log_path_cust, aptr->log_path);
    strcat(err_log_path_cust, "customer.err");
    rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust, DB_IN);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(c_hdbc1);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 30000));
        rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
    }

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

    rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
    strcpy(err_log_path_hist, aptr->log_path);
    strcat(err_log_path_hist, "history.err");
    rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    sprintf(bcphint, "tablock");
    rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

    customer_rows_loaded = 0;
    history_rows_loaded = 0;

    CustomerBufInit();

    customer_time_start.time_start = (TimeNow() / MILLI);
    history_time_start.time_start = (TimeNow() / MILLI);

    for (w_id = (long)aptr->starting_warehouse; w_id <= 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 = %d, w_id
= %d\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
= %d\n", d_id, w_id);
                                hThread[1] = CreateThread(NULL,
0,
(LPTHREAD_START_ROUTINE) LoadHistoryTable,
&history_time_start,
0,
&dwThreadID[1]);
                                if (hThread[1] == NULL)
                                {
                                        printf("Error, failed in creating creating
thread = 1.\n");
                                        exit(-1);
                                }
                                WaitForSingleObject( hThread[0], INFINITE );
                                WaitForSingleObject( hThread[1], INFINITE );
                                if (CloseHandle(hThread[0]) == FALSE)
                                {
                                        printf("Error, failed in closing customer
thread handle with errno: %d\n", GetLastError());
                                }
                                if (CloseHandle(hThread[1]) == FALSE)
                                {
                                        printf("Error, failed in closing history
thread handle with errno: %d\n", GetLastError());
                                }
                                }
                                // flush the bulk connection
                                rcint = bcp_done(c_hdbc1);
                                if (rcint < 0)
                                        HandleErrorDBC(c_hdbc1);
                                rcint = bcp_done(c_hdbc2);
                                if (rcint < 0)
                                        HandleErrorDBC(c_hdbc2);
                                printf("Finished loading customer table.\n");
                                // if build index after load...
                                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                {
                                        BuildIndex("idxcuscl");
                                        // check the number of processors on this system

```

```

                                // if 8 or more processors, then build index on History.
                                // if less than 8 processors, do not build the index
                                num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
                                if (num_procs >= 8)
                                        BuildIndex("idxhiscl");
                                }
                                // build non-clustered index
                                if (aptr->build_index == 1)
                                        BuildIndex("idxcusnc");
                                // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
                                // C_W_ID = 1, and C_D_ID = 1
                                sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first
= 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" > %snurand_load.log",
aptr->server,
aptr->user,
aptr->password,
aptr->database,
LOADER_NURAND_C,
aptr->log_path);
                                system(cmd);
                                SQLFreeStmt(c_hstmt1, SQL_DROP);
                                SQLDisconnect(c_hdbc1);
                                SQLFreeConnect(c_hdbc1);
                                SQLFreeStmt(c_hstmt2, SQL_DROP);
                                SQLDisconnect(c_hdbc2);
                                SQLFreeConnect(c_hdbc2);
                                return;
                                }
                                //=====
                                //
                                // Function : CustomerBufInit
                                //
                                //=====
                                void CustomerBufInit()
                                {
                                        long i;
                                        for (i=0;i<customers_per_district;i++)
                                        {
                                                customer_buf[i].c_id = 0;
                                                customer_buf[i].c_d_id = 0;
                                                customer_buf[i].c_w_id = 0;
                                                strcpy(customer_buf[i].c_first,"");
                                                strcpy(customer_buf[i].c_middle,"");
                                                strcpy(customer_buf[i].c_last,"");
                                                strcpy(customer_buf[i].c_street_1,"");
                                                strcpy(customer_buf[i].c_street_2,"");
                                                strcpy(customer_buf[i].c_city,"");
                                                strcpy(customer_buf[i].c_state,"");
                                                strcpy(customer_buf[i].c_zip,"");
                                                strcpy(customer_buf[i].c_phone,"");
                                                strcpy(customer_buf[i].c_credit,"");

```

```

customer_buf[i].c_credit_lim = 0;
customer_buf[i].c_discount = (float) 0;

strcpy(customer_buf[i].c_balance, "");

customer_buf[i].c_ytd_payment = 0;
customer_buf[i].c_payment_cnt = 0;
customer_buf[i].c_delivery_cnt = 0;

strcpy(customer_buf[i].c_data, "");

customer_buf[i].h_amount = 0;

strcpy(customer_buf[i].h_data, "");
}
}

//=====
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//=====
void CustomerBufLoad(int d_id, long w_id)
{
    long          i;
    CUSTOMER_SORT_STRUCT  c[CUSTOMERS_PER_DISTRICT];

    for (i=0;i<customers_per_district;i++)
    {
        if (i < 1000)
            LastName(i, c[i].c_last);
        else
            LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);

        MakeAlphaStringPadded(8,16,FIRST_NAME_LEN, c[i].c_first);

        c[i].c_id = i+1;
    }

    printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
d_id, w_id);

    for (i=0;i<customers_per_district;i++)
    {
        customer_buf[i].c_d_id = d_id;
        customer_buf[i].c_w_id = w_id;
        customer_buf[i].h_amount = 10.0;
        customer_buf[i].c_ytd_payment = 10.0;
        customer_buf[i].c_payment_cnt = 1;
        customer_buf[i].c_delivery_cnt = 0;
        customer_buf[i].c_id = c[i].c_id;
        strcpy(customer_buf[i].c_first, c[i].c_first);
        strcpy(customer_buf[i].c_last, c[i].c_last);
        customer_buf[i].c_middle[0] = 'O';
        customer_buf[i].c_middle[1] = 'E';
        MakeAddress(customer_buf[i].c_street_1,
customer_buf[i].c_street_2,
customer_buf[i].c_city,
customer_buf[i].c_state,

```

```

customer_buf[i].c_zip);
MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

if (RandomNumber(1L, 100L) > 10)
    customer_buf[i].c_credit[0] = 'G';
else
    customer_buf[i].c_credit[0] = 'B';
customer_buf[i].c_credit[1] = 'C';
customer_buf[i].c_credit_lim = 50000.0;
customer_buf[i].c_discount = ((float) RandomNumber(0L, 5000L)) /
10000.0;

strcpy(customer_buf[i].c_balance, "-10.0");
MakeAlphaStringPadded(300, 500, C_DATA_LEN,
customer_buf[i].c_data);

// Generate HISTORY data
MakeAlphaStringPadded(12, 24, H_DATA_LEN,
customer_buf[i].h_data);
}
}

//=====
//
// Function   : LoadCustomerTable
//
//=====
void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
    long          i;
    long          c_id;
    short         c_d_id;
    long          c_w_id;
    char          c_first[FIRST_NAME_LEN+1];
    char          c_middle[MIDDLE_NAME_LEN+1];
    char          c_last[LAST_NAME_LEN+1];
    char          c_street_1[ADDRESS_LEN+1];
    char          c_street_2[ADDRESS_LEN+1];
    char          c_city[ADDRESS_LEN+1];
    char          c_state[STATE_LEN+1];
    char          c_zip[ZIP_LEN+1];
    char          c_phone[PHONE_LEN+1];
    char          c_credit[CREDIT_LEN+1];
    double        c_credit_lim;
    double        c_discount;
    char          c_balance[6];
    double        c_ytd_payment;
    short         c_payment_cnt;
    short         c_delivery_cnt;
    char          c_data[C_DATA_LEN+1];
    char          c_since[C_SINCE_LEN+1];
    RETCODE       rc;

    i = 0;
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != 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 != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0,
SQLCHARACTER, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN, NULL, 0, 0,
++i);

```

```

        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);
        rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, C_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc1);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id = customer_buf[i].c_id;
            c_d_id = customer_buf[i].c_d_id;
            c_w_id = customer_buf[i].c_w_id;

            strcpy(c_first, customer_buf[i].c_first);
            strcpy(c_middle, customer_buf[i].c_middle);
            strcpy(c_last, customer_buf[i].c_last);
            strcpy(c_street_1, customer_buf[i].c_street_1);
            strcpy(c_street_2, customer_buf[i].c_street_2);
            strcpy(c_city, customer_buf[i].c_city);
            strcpy(c_state, customer_buf[i].c_state);
            strcpy(c_zip, customer_buf[i].c_zip);
            strcpy(c_phone, customer_buf[i].c_phone);
            strcpy(c_credit, customer_buf[i].c_credit);

            FormatDate(&c_since);

            c_credit_lim = customer_buf[i].c_credit_lim;
            c_discount = customer_buf[i].c_discount;
            strcpy(c_balance, customer_buf[i].c_balance);
            c_ytd_payment = customer_buf[i].c_ytd_payment;
            c_payment_cnt = customer_buf[i].c_payment_cnt;
            c_delivery_cnt = customer_buf[i].c_delivery_cnt;
            strcpy(c_data, customer_buf[i].c_data);

            // Send data to server
            rc = bcp_sendrow(c_hdbc1);
            if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

            customer_rows_loaded++;
            CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded,
"customer", &customer_time_start->time_start);
        }
    }

//=====
//
// Function   : LoadHistoryTable
//
//=====
void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
    long        c_id;           i;
    short       c_d_id;
    long        c_w_id;
    double      h_amount;
    char        h_data[H_DATA_LEN+1];
    char        h_date[H_DATE_LEN+1];
    RETCODE     rc;

```

```

        i = 0;
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0,
SQLCHARACTER, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
            HandleErrorDBC(c_hdbc2);

        for (i = 0; i < customers_per_district; i++)
        {
            c_id = customer_buf[i].c_id;
            c_d_id = customer_buf[i].c_d_id;
            c_w_id = customer_buf[i].c_w_id;
            h_amount = customer_buf[i].h_amount;
            strcpy(h_data, customer_buf[i].h_data);

            FormatDate(&h_date);

            // send to server
            rc = bcp_sendrow(c_hdbc2);
            if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

            history_rows_loaded++;
            CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded,
"history", &history_time_start->time_start);
        }
    }

//=====
//
// Function   : LoadOrders
//
//=====
void LoadOrders()
{

```

```

        LOADER_TIME_STRUCT    orders_time_start;
        LOADER_TIME_STRUCT    new_order_time_start;
        LOADER_TIME_STRUCT    order_line_time_start;
        long                   w_id;
        short                  d_id;
        DWORD                  dwThreadID[MAX_ORDER_THREADS];
        HANDLE                 hThread[MAX_ORDER_THREADS];
        char                   name[20];
        RETCODE                 rc;
        char                   bcphint[128];
        char                   err_log_path_ord[256];
        char                   err_log_path_nord[256];
        char                   err_log_path_ordl[256];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
            BuildIndex("idxordcl");
            BuildIndex("idxnodcl");
            BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s.%s", aptr->database, "orders");

        rc = bcp_init(o_hdbc1, name, NULL, "logs\\orders.err", DB_IN);
        strcpy(err_log_path_ord, aptr->log_path);
        strcat(err_log_path_ord, "orders.err");
        rc = bcp_init(o_hdbc1, name, NULL, err_log_path_ord, DB_IN);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
            sprintf(bcphint, "tablock, order (o_w_id, o_d_id, o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 3000));
            rc = bcp_control(o_hdbc1, BCPHINTS, (void*) bcphint);
            if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);
        }

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

        rc = bcp_init(o_hdbc2, name, NULL, "logs\\neword.err", DB_IN);
        strcpy(err_log_path_nord, aptr->log_path);
        strcat(err_log_path_nord, "neword.err");
        rc = bcp_init(o_hdbc2, name, NULL, err_log_path_nord, DB_IN);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
            sprintf(bcphint, "tablock, order (no_w_id, no_d_id, no_o_id),
ROWS_PER_BATCH = %u", (aptr->num_warehouses * 9000));
            rc = bcp_control(o_hdbc2, BCPHINTS, (void*) bcphint);
            if (rc != SUCCEED)

```

```

        HandleErrorDBC(o_hdbc2);
    }

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

    rc = bcp_init(o_hdbc3, name, NULL, "logs\\ordline.err", DB_IN);
    strcpy(err_log_path_ordl, aptr->log_path);
    strcat(err_log_path_ordl, "ordline.err");
    rc = bcp_init(o_hdbc3, name, NULL, err_log_path_ordl, DB_IN);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    if ((aptr->build_index == 1) && (aptr->index_order == 1))
    {
        sprintf(bcphint, "tablock, order (ol_w_id, ol_d_id, ol_o_id,
ol_number), ROWS_PER_BATCH = %u", (aptr->num_warehouses * 300000));
        rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc3);
    }

    orders_rows_loaded      = 0;
    new_order_rows_loaded  = 0;
    order_line_rows_loaded = 0;

    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 <= 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 = %d,
w_id = %d\n", d_id, w_id);

    hThread[2] = CreateThread(NULL,

0,

(LPTHREAD_START_ROUTINE) LoadOrderLineTable,

&order_line_time_start,

0,

&dwThreadID[2]);

    if (hThread[2] == NULL)
    {
        printf("Error, failed in creating creating
thread = 2.\n");
        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 = %d, w_id = %d\n",
           d_id, w_id);

    GetPermutation(cust, orders_per_district);

    for (o_id=0;o_id<orders_per_district;o_id++)
    {
        // Generate ORDER and NEW-ORDER data
        orders_buf[o_id].o_d_id = d_id;
        orders_buf[o_id].o_w_id = w_id;
        orders_buf[o_id].o_id = o_id+1;
        orders_buf[o_id].o_c_id = cust[o_id+1];
        orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

        if (o_id < first_new_order)
        {
            orders_buf[o_id].o_carrier_id =
(short)RandomNumber(1L, 10L);
            orders_buf[o_id].o_all_local = 1;
        }
        else
        {
            orders_buf[o_id].o_carrier_id = 0;
            orders_buf[o_id].o_all_local = 1;
        }

        for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
        {
            orders_buf[o_id].o_ol[ol].ol = ol+1;
            orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L,
max_items);
            orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
            orders_buf[o_id].o_ol[ol].ol_quantity = 5;
            MakeAlphaString(24, 24, OL_DIST_INFO_LEN,
&orders_buf[o_id].o_ol[ol].ol_dist_info);

            // Generate ORDER-LINE data
            if (o_id < first_new_order)
            {
                orders_buf[o_id].o_ol[ol].ol_amount = 0;
                // Added to insure ol_delivery_d set
                properly during load

                FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);
            }
            else
            {
                orders_buf[o_id].o_ol[ol].ol_amount =
RandomNumber(1,999999)/100.0;
                // Added to insure ol_delivery_d set
                properly during load
                // odbc datetime format

                strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");
            }
        }
    }
}

```

```

    }
}

//=====
//
// Function   : LoadOrdersTable
//
//=====
void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
    int         i;
    long        o_id;
    short       o_d_id;
    long        o_w_id;

    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;

    char        o_entry_d[O_ENTRY_D_LEN+1];
    RETCODE     rc;
    DBINT       rcint;

    // bind ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0,
SQLCHARACTER, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id         = orders_buf[i].o_id;

```

```

        o_d_id      = orders_buf[i].o_d_id;
        o_w_id      = orders_buf[i].o_w_id;
        o_c_id      = orders_buf[i].o_c_id;
        o_carrier_id = orders_buf[i].o_carrier_id;
        o_ol_cnt    = orders_buf[i].o_ol_cnt;
        o_all_local = orders_buf[i].o_all_local;

        FormatDate(&o_entry_d);

        // send data to server
        rc = bcp_sendrow(o_hdbc1);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);

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

    if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc1);
        if (rcint < 0)
            HandleErrorDBC(o_hdbc1);

        SQLFreeStmt(o_hstmt1, SQL_DROP);
        SQLDisconnect(o_hdbc1);
        SQLFreeConnect(o_hdbc1);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
            BuildIndex("idxordc1");

        // build non-clustered index
        if (aptr->build_index == 1)
            BuildIndex("idxordnc");
    }
}

//=====
//
// Function   : LoadNewOrderTable
//
//=====
void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
    long        i;
    long        o_id;
    short       o_d_id;
    long        o_w_id;
    RETCODE     rc;
    DBINT       rcint;

    // Bind NEW-ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);

```

```

    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc2);

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

        rc = bcp_sendrow(o_hdbc2);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;
        CheckForCommit_Big(o_hdbc2, o_hstmt2, new_order_rows_loaded,
"new_order", &new_order_time_start->time_start);
    }

    if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
    {
        rcint = bcp_done(o_hdbc2);
        if (rcint < 0)
            HandleErrorDBC(o_hdbc2);

        SQLFreeStmt(o_hstmt2, SQL_DROP);
        SQLDisconnect(o_hdbc2);
        SQLFreeConnect(o_hdbc2);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
            BuildIndex("idxnodc1");
    }
}

//=====
//
// Function : LoadOrderLineTable
//
//=====
void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
    long i;
    long j;
    long o_id;
    short o_d_id;
    long o_w_id;
    double ol;
    long ol_i_id;
    long ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
    char RETCODE rc;
    DBINT rcint;

    // bind ORDER-LINE data

```

```

    i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN,
NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0,
SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL,
0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0,
++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;

        for (j=0; j < orders_buf[i].o_ol_cnt; j++)
        {
            ol = orders_buf[i].o_ol[j].ol;
            ol_i_id = orders_buf[i].o_ol[j].ol_i_id;
            ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
            ol_quantity = orders_buf[i].o_ol[j].ol_quantity;
            ol_amount = orders_buf[i].o_ol[j].ol_amount;

            strcpy(ol_delivery_d, orders_buf[i].o_ol[j].ol_delivery_d);

            strcpy(ol_dist_info, orders_buf[i].o_ol[j].ol_dist_info);

```

```

rc = bcp_sendrow(o_hdbc3);
if (rc != 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 == aptr->num_warehouses) && (o_d_id == 10))
{
    rcint = bcp_done(o_hdbc3);
    if (rcint < 0)
        HandleErrorDBC(o_hdbc3);

    SQLFreeStmt(o_hstmt3, SQL_DROP);
    SQLDisconnect(o_hdbc3);
    SQLFreeConnect(o_hdbc3);

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr->index_order == 0))
        BuildIndex("idxodlcl");
}

}

//=====
//
// Function : GetPermutation
//
//=====
void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1;i<=n;i++)
        perm[i] = i;

    for (i=1;i<=n;i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

//=====
//
// Function : CheckForCommit
//
//=====
void CheckForCommit(HDBC hdbc,
                    HSTMT hstmt,
                    long rows_loaded,
                    char *table_name,
                    long *time_start)
{

```

```

long time_end, time_diff;

if ( !(rows_loaded % aptr->batch) )
{
    time_end = (TimeNow() / MILLI);
    time_diff = time_end - *time_start;

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

    *time_start = time_end;
}

return;
}

//=====
//
// Function : CheckForCommit_Big
//
//=====
void CheckForCommit_Big(HDBC hdbc,
                       HSTMT 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_diff = time_end - *time_start;

        printf("-> Loaded %ld rows into %s in %ld sec - Total = %.0f
(%.2f rps)\n",
               aptr->batch,
               table_name,
               time_diff,
               rows_loaded,
               (float) aptr->batch / (time_diff ? time_diff
: 1L));

        *time_start = time_end;
    }

    return;
}

//=====
//
// Function : OpenConnections
//
//=====

```

```

void OpenConnections()
{
    RETCODE          rc;

    char              szDriverString[300];
    char              szDriverStringOut[1024];
    SQLSMALLINT       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(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );
    SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON,
SQL_IS_INTEGER );

    // Open connections to SQL Server
    // Connection 1
    sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                                    aptr->server,
                                                    aptr->user,
                                                    aptr->password,
                                                    aptr->database );

    rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = SQLDriverConnect ( i_hdbc1,

                                                                    NULL,

        (SQLCHAR*)&szDriverString[0] ,

                                                                    SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

                                                                    NULL,

        sizeof(szDriverStringOut),

                                                                    &cbDriverStringOut,
                                                                    SQL_DRIVER_NOPROMPT );
}

```

```

if ( (rc != SUCCEED) &&
      (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(i_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 2
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                                    aptr->server,
                                                    aptr->user,
                                                    aptr->password,
                                                    aptr->database );

rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);

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

rc = SQLDriverConnect ( w_hdbc1,

                                                                    NULL,

        (SQLCHAR*)&szDriverString[0] ,

                                                                    SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

                                                                    NULL,

        sizeof(szDriverStringOut),

                                                                    &cbDriverStringOut,
                                                                    SQL_DRIVER_NOPROMPT );

if ( (rc != SUCCEED) &&
      (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(w_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 3
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

                                                    aptr->server,
                                                    aptr->user,
                                                    aptr->password,
                                                    aptr->database );

rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,

                                                                    NULL,

        (SQLCHAR*)&szDriverString[0] ,

                                                                    SQL_NTS,

```



```

(SQLCHAR*)&szDriverStringOut[0],
sizeof(szDriverStringOut),
&cbDriverStringOut,
SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 4
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc2);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 5
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc1);

```

```

rc = SQLDriverConnect ( o_hdbc1,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 6
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc2);

rc = SQLDriverConnect ( o_hdbc2,
                                NULL,
                                (SQLCHAR*)&szDriverString[0] ,
                                SQL_NTS,
                                (SQLCHAR*)&szDriverStringOut[0],
                                sizeof(szDriverStringOut),
                                &cbDriverStringOut,
                                SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc2);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 7
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->database );

```

```

rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
    HandleErrorDBC(o_hdbc3);

rc = SQLDriverConnect ( o_hdbc3,
                        NULL,
                        (SQLCHAR*)&szDriverString[0] ,
                        SQL_NTS,
                        (SQLCHAR*)&szDriverStringOut[0],
                        sizeof(szDriverStringOut),
                        &cbDriverStringOut,
                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
     (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(o_hdbc3);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

//=====
//
// Function name: BuildIndex
//
//=====
void BuildIndex(char      *index_script)
{
    char      cmd[256];

    printf("Starting index creation:  %s\n",index_script);

    sprintf(cmd, "osql -S%s -U%s -P%s -e -i%s\\%s.sql > %s%s.log",
            aptr->server,
            aptr->user,
            aptr->password,
            aptr->index_script_path,
            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];

```

```

SQLLEN      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,"tpcldr.err");
        fp1 = fopen(err_log_path,"a+");
        if (fp1 == NULL)
            printf("ERROR:  Unable to open errorlog file.\n");
        else
            fprintf(fp1, "[%s : %s] %s\nSQLState: %s\n" , datebuf,
timebuf, szLastError, SqlState);
        fclose(fp1);
    }
    i++;
}

//=====
//
// Function : HandleErrorSTMT
//
//=====
void HandleErrorSTMT (HSTMT  hstmt1)
{
    SQLCHAR      SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
    SQLLEN      NativeError;
    SQLSMALLINT i, MsgLen;
    SQLRETURN   rc2;
    char        timebuf[128];
    char        datebuf[128];
    char        err_log_path[256];
    FILE        *fp1;

    i = 1;
    while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState ,
&NativeError,
                                Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
    {
        if (total_db_errors >= MAX_SQL_ERRORS)

```

```

        {
            printf(">>>> Maximum SQL errors of %d exceeded.
Terminating TPCCLDR.<<<<\n",total_db_errors);
            exit(9);
        }
        total_db_errors++;

        sprintf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\nSQLState: %s\n" , datebuf, timebuf,
szLastError, SqlState);

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"tpcldr.err");
        fpl = fopen(err_log_path,"a+");
        if (fpl == NULL)
            printf("ERROR:  Unable to open errorlog file.\n");
        else
        {
            fprintf(fpl, "[%s : %s] %s\nSQLState: %s\n" , datebuf,
timebuf, szLastError, SqlState);
            fclose(fpl);
        }

        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;
}

```

Appendix C: Tunable Parameters

Microsoft SQL Server 2005 Installation Procedures

Microsoft SQL Server 2005 Enterprise x64 Edition SP2
Installation Procedures
Type of installation: custom
During the custom installation, use the default
settings for all except the following two areas:
Services accounts:
SQL Server - local system account
SQL Server Agent - local system account
Set the sort order/collation as Latin1_General / BIN

Microsoft SQL Server Configuration Parameters

```

1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> -----
-----
--
-- File:  VERSION.SQL
--
--      Microsoft TPC-C Benchmark Kit Ver. 4.68
--
--      Copyright Microsoft, 2006
--
--
--      - Extracts current version of SQL Server
--
-----
USE master
1> 2> 3> 4> 5>

```

```

SELECT CONVERT(char(20),
SERVERPROPERTY('ProductVersion')),
CONVERT(char(20),
SERVERPROPERTY('ProductLevel')),
CONVERT(char(29), SERVERPROPERTY('Edition'))

-----
-----
9.00.3042.00      SP2      Enterprise
Edition (64-bit)

(1 row affected)
1> 2> 3>
SELECT CONVERT(char(30), GETDATE(), 21)

-----
-----
2009-11-04 15:42:41.920

(1 row affected)
1>
1> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14>
-----
-----
--
-- File:  CONFIG.SQL
--
--      Microsoft TPC-C Benchmark Kit Ver. 4.68
--
--      Copyright Microsoft, 2006
--
--      - Collects SQL Server configuration
parameters      --
--
-----
-----
PRINT ' '
SELECT CONVERT(char(30), GETDATE(), 21)
PRINT ' '

-----
-----
2009-11-04 15:42:42.077

(1 row affected)

1> 2> 3> Configuration option 'show advanced options'
changed from 1 to 1. Run the RECONFIGURE statement to
install.

sp_configure 'show advanced',1
1> 2> 3>
RECONFIGURE WITH OVERRIDE
1> 2> 3>
sp_configure
name      minimum
maximum  config_value run_value
-----
-----

```

```

Ad Hoc Distributed Queries      0
1      0      0
affinity I/O mask      -2147483648
2147483647      0      0
affinity mask      -2147483648
2147483647      65535      65535
affinity64 I/O mask      -2147483648
2147483647      0      0
affinity64 mask      -2147483648
2147483647      0      0
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      1      1
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      704      704
lightweight pooling      0
1      1      1
locks      5000
2147483647      0      0
max degree of parallelism      0
64      1      1
max full-text crawl range      0
256      4      4

```

```

max server memory (MB)          32767 141312 141312 16
2147483647 141312 141312 0
max text repl size (B)         32767 65536 65536 0
2147483647 65536 65536 128
max worker threads             32767 1800 1800 0
media retention                 365 0 0 0
min memory per query (KB)      2147483647 512 512 512
2147483647 512 512 0
min server memory (MB)         2147483647 0 0 0
nested triggers                1 1 1 0
network packet size (B)        32767 2048 2048 512
Ole Automation Procedures      1 0 0 0
open objects                   2147483647 0 0 0
PH timeout (s)                 3600 60 60 1
precompute rank                1 0 0 0
priority boost                  1 1 1 0
query governor cost limit      2147483647 0 0 0
query wait (s)                 2147483647 -1 -1 -1
recovery interval (min)        32767 32767 32767 0
remote access                   1 1 1 0
remote admin connections       1 0 0 0
remote login timeout (s)       2147483647 20 20 0
remote proc trans              1 0 0 0
remote query timeout (s)       2147483647 600 600 0
Replication XPs                1 0 0 0
scan for startup procs         1 0 0 0
server trigger recursion       1 1 1 0
set working set size           1 0 0 0
show advanced options          1 1 1 0
SMO and DMO XPs                1 1 1 0
SQL Mail XPs                   1 0 0 0
transform noise words          1 0 0 0
two digit year cutoff          9999 2049 2049 1753
user connections               32767 0 0 0

```

```

user options                    32767 0 0 0
Web Assistant Procedures        1 0 0 0
xp_cmdshell                     1 0 0 0
1>

```

Microsoft SQL Server Node Configuration Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node0
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM
Value 0
Name: CPUMask
Type: REG_DWORD
Data: 0xff

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\90\NodeConfiguration\Node1
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM
Value 0
Name: CPUMask
Type: REG_DWORD
Data: 0xff00

```

Microsoft SQL Server Super Socket

Configuration Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM
Value 0
Name: Certificate
Type: REG_SZ
Data:

```

```

Value 1
Name: DisplayName
Type: REG_SZ
Data: SQL Server Network Configuration

```

```

Value 2
Name: ForceEncryption
Type: REG_DWORD
Data: 0

```

```

Value 3
Name: HideInstance
Type: REG_DWORD
Data: 0

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\AdminCon
nection
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM
Value 0
Name: DisplayName
Type: REG_SZ
Data: Dedicated Administrative
Connection

```

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\AdminCon
nection\Tcp
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM
Value 0
Name: DisplayName
Type: REG_SZ
Data: TCP/IP

```

```

Value 1
Name: TcpDynamicPorts
Type: REG_SZ
Data: 1434

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Np
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM

Value 0
Name: DisplayName
Type: REG_SZ
Data: Named Pipes

Value 1
Name: Enabled
Type: REG_DWORD
Data: 0

Value 2
Name: PipeName
Type: REG_SZ
Data: \\.\pipe\sql\query

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Sm
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM

Value 0
Name: DisplayName
Type: REG_SZ
Data: Shared Memory

Value 1
Name: Enabled
Type: REG_DWORD
Data: 0x1

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:48 PM

Value 0
Name: DisplayName
Type: REG_SZ
Data: TCP/IP

Value 1
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 2
Name: KeepAlive
Type: REG_DWORD
Data: 0x7530

Value 3
Name: ListenOnAllIPs
Type: REG_DWORD
Data: 0x1

Value 4
Name: NoDelay
Type: REG_DWORD
Data: 0

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP1
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:18 AM

Value 0
Name: Active
Type: REG_DWORD
Data: 0x1

Value 1
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 2
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 3
Name: IPAddress
Type: REG_SZ
Data: 130.132.208.3

Value 4
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 5
Name: TcpPort
Type: REG_SZ
Data: 2004

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP2
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:18 AM

Value 0
Name: Active
Type: REG_DWORD
Data: 0x1

Value 1
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 2
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 3
Name: IPAddress
Type: REG_SZ
Data: 130.131.208.2

Value 4
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 5
Name: TcpPort
Type: REG_SZ
Data: 2003

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP3
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:18 AM

Value 0
Name: Active
Type: REG_DWORD
Data: 0x1

Value 1
Name: DisplayName
Type: REG_SZ
Data: Specific IP Address

Value 2
Name: Enabled
Type: REG_DWORD
Data: 0x1

Value 3
Name: IPAddress
Type: REG_SZ
Data: 130.130.208.1

Value 4
Name: TcpDynamicPorts
Type: REG_SZ
Data:

Value 5
Name: TcpPort
Type: REG_SZ
Data: 2002

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP4
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:18 AM

Value 0
Name: Active
Type: REG_DWORD
Data: 0x1

Value 1
 Name: DisplayName
 Type: REG_SZ
 Data: Specific IP Address

Value 2
 Name: Enabled
 Type: REG_DWORD
 Data: 0x1

Value 3
 Name: IpAddress
 Type: REG_SZ
 Data: 130.168.208.20

Value 4
 Name: TcpDynamicPorts
 Type: REG_SZ
 Data:

Value 5
 Name: TcpPort
 Type: REG_SZ
 Data: 2001

Key Name:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
 Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IP5
 Class Name: <NO CLASS>
 Last Write Time: 10/27/2009 - 4:48 PM

Value 0
 Name: Active
 Type: REG_DWORD
 Data: 0x1

Value 1
 Name: DisplayName
 Type: REG_SZ
 Data: Specific IP Address

Value 2
 Name: Enabled
 Type: REG_DWORD
 Data: 0

Value 3
 Name: IpAddress
 Type: REG_SZ
 Data: 127.0.0.1

Value 4
 Name: TcpDynamicPorts
 Type: REG_SZ
 Data:

Value 5
 Name: TcpPort
 Type: REG_SZ
 Data: 1433

Key Name:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
 Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IPAL
 1
 Class Name: <NO CLASS>
 Last Write Time: 10/27/2009 - 4:48 PM

Value 0
 Name: DisplayName
 Type: REG_SZ
 Data: Any IP Address

Value 1
 Name: TcpDynamicPorts
 Type: REG_SZ
 Data:

Value 2
 Name: TcpPort
 Type: REG_SZ
 Data: 2001[0x1],2002[0x1],2003[0x2],2004[0x2]

Key Name:
 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL
 Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Via
 Class Name: <NO CLASS>
 Last Write Time: 10/27/2009 - 4:48 PM

Value 0
 Name: DefaultServerPort
 Type: REG_SZ
 Data: 0:1433

Value 1
 Name: DisplayName
 Type: REG_SZ
 Data: VIA

Value 2
 Name: Enabled
 Type: REG_DWORD
 Data: 0

Value 3
 Name: ListenInfo
 Type: REG_SZ
 Data: 0:1433

Database Server System Configuration

System Information report written at: 11/03/09
 11:55:20
 System Name: HOOPS
 [System Summary]

Item Value
 OS Name Microsoft® Windows Server® 2008 Enterprise
 Version 6.0.6002 Service Pack 2, v.286 Build 6002

Other OS Description Not Available
 OS Manufacturer Microsoft Corporation
 System Name HOOPS
 System Manufacturer HP
 System Model ProLiant DL370 G6
 System Type x64-based PC
 Processor Intel(R) Xeon(R) CPU W5580 @
 3.20GHz, 3199 Mhz, 4 Core(s), 8 Logical Processor(s)

Processor Intel(R) Xeon(R) CPU W5580 @
 3.20GHz, 3199 Mhz, 4 Core(s), 8 Logical Processor(s)

BIOS Version/Date HP P63, 10/1/2009
 SMBIOS Version 2.6
 Windows Directory C:\Windows
 System Directory C:\Windows\system32
 Boot Device \Device\HarddiskVolume66

Locale United States
 Hardware Abstraction Layer Version =
 "6.0.6002.16670"
 User Name HOOPS\Administrator
 Time Zone Central Standard Time
 Installed Physical Memory (RAM) 144 GB
 Total Physical Memory 144 GB
 Available Physical Memory 136 GB
 Total Virtual Memory 148 GB
 Available Virtual Memory 144 GB
 Page File Space 8.47 GB
 Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
IRQ 30	PCI standard PCI-to-PCI bridge
IRQ 30	PCI standard PCI-to-PCI bridge
IRQ 30	PCI standard PCI-to-PCI bridge
IRQ 30	Smart Array P411 Controller (Non-Miniport)

Memory Address 0xF5100000-0xF52FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address 0xF5100000-0xF52FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address 0xF5100000-0xF52FFFFFF	PCI standard PCI-to-PCI bridge

I/O Port 0x00000000-0x0000003F	PCI bus
I/O Port 0x00000000-0x0000003F	Direct memory access controller

IRQ 20	Standard Universal PCI to USB Host Controller
--------	---

IRQ 20 Standard Enhanced PCI to USB Host Controller

I/O Port 0x000003C0-0x000003DF PCI bus
I/O Port 0x000003C0-0x000003DF Standard VGA Graphics Adapter

I/O Port 0x00000070-0x00000077 Motherboard resources
I/O Port 0x00000070-0x00000077 System CMOS/real time clock

IRQ 10 Base System Device
IRQ 10 Base System Device
IRQ 10 IPMI Interface

Memory Address 0xE8000000-0xEFFFFFFF Intel(R) 82801 PCI Bridge - 244E
Memory Address 0xE8000000-0xEFFFFFFF Standard VGA Graphics Adapter

IRQ 22 Standard Universal PCI to USB Host Controller
IRQ 22 Standard Universal PCI to USB Host Controller

IRQ 23 Standard Universal PCI to USB Host Controller
IRQ 23 Standard Universal PCI to USB Host Controller

I/O Port 0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge
I/O Port 0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge
I/O Port 0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge

I/O Port 0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge
I/O Port 0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge
I/O Port 0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge

IRQ 37 PCI standard PCI-to-PCI bridge
IRQ 37 Smart Array P411 Controller (Non-Miniport)

IRQ 37 PCI standard PCI-to-PCI bridge
IRQ 37 Smart Array P411 Controller (Non-Miniport)

IRQ 37 PCI standard PCI-to-PCI bridge
IRQ 37 Smart Array P411 Controller (Non-Miniport)

Memory Address 0xFED00000-0xFED03FFF PCI bus
Memory Address 0xFED00000-0xFED03FFF PCI bus
Memory Address 0xFED00000-0xFED03FFF High precision event timer

IRQ 16 PCI standard PCI-to-PCI bridge

IRQ 16 Smart Array P411 Controller (Non-Miniport)

IRQ 39 PCI standard PCI-to-PCI bridge
IRQ 39 Smart Array P411 Controller (Non-Miniport)

IRQ 39 PCI standard PCI-to-PCI bridge
IRQ 39 Smart Array P411 Controller (Non-Miniport)

Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF Standard VGA Graphics Adapter

Memory Address 0xF5B00000-0xF87FFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF5B00000-0xF87FFFFF PCI standard PCI-to-PCI bridge

I/O Port 0x000003B0-0x000003BB PCI bus
I/O Port 0x000003B0-0x000003BB Standard VGA Graphics Adapter

I/O Port 0x00001000-0x0000FFFF PCI bus
I/O Port 0x00001000-0x0000FFFF Standard Universal PCI to USB Host Controller

[DMA]

Resource Device Status
Channel 7 Direct memory access controller OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource Device Status
0x00001000-0x0000FFFF PCI bus OK
0x00001000-0x0000FFFF Standard Universal PCI to USB Host Controller OK
0x00000000-0x000003AF PCI bus OK
0x00000000-0x000003AF Direct memory access controller OK
0x000003E0-0x00000CF7 PCI bus OK
0x0000D000-0x0000FFFF 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
0x00008000-0x00008FFF PCI standard PCI-to-PCI bridge OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK
0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK

0x00006000-0x00007FFF PCI standard PCI-to-PCI bridge OK
0x00007000-0x00007FFF PCI standard PCI-to-PCI bridge OK
0x0000F000-0x0000FFFF PCI standard PCI-to-PCI bridge OK
0x00005000-0x00005FFF PCI standard PCI-to-PCI bridge OK
0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge OK
0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge OK
0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge OK
0x00009000-0x0000EFFF PCI standard PCI-to-PCI bridge OK
0x0000A000-0x0000AFFF PCI standard PCI-to-PCI bridge OK
0x0000B000-0x0000BFFF PCI standard PCI-to-PCI bridge OK
0x0000C000-0x0000CFFF PCI standard PCI-to-PCI bridge OK
0x0000D000-0x0000DFFF PCI standard PCI-to-PCI bridge OK
0x0000E000-0x0000EFFF PCI standard PCI-to-PCI bridge OK
0x00004000-0x00004FFF PCI standard PCI-to-PCI bridge OK
0x00001020-0x0000103F Standard Universal PCI to USB Host Controller OK
0x00001040-0x0000105F Standard Universal PCI to USB Host Controller OK
0x00001060-0x0000107F Standard Universal PCI to USB Host Controller OK
0x00002000-0x00003FFF Intel(R) 82801 PCI Bridge - 244E OK
0x00003000-0x000030FF Standard VGA Graphics Adapter OK
0x00002800-0x000028FF Base System Device OK

0x00003400-0x000034FF Base System Device OK

0x00003800-0x0000381F Standard Universal PCI to USB Host Controller OK
0x00000070-0x00000077 Motherboard resources OK
0x00000070-0x00000077 System CMOS/real time clock 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 OK	Motherboard resources	IRQ 81	Microsoft ACPI-Compliant System	OK	IRQ 112	Microsoft ACPI-Compliant System	OK
0x00000010-0x0000001F OK	Motherboard resources	IRQ 82	Microsoft ACPI-Compliant System	OK	IRQ 113	Microsoft ACPI-Compliant System	OK
0x00000C80-0x00000C83 OK	Motherboard resources	IRQ 83	Microsoft ACPI-Compliant System	OK	IRQ 114	Microsoft ACPI-Compliant System	OK
0x00000CD4-0x00000CD7 OK	Motherboard resources	IRQ 84	Microsoft ACPI-Compliant System	OK	IRQ 115	Microsoft ACPI-Compliant System	OK
0x00000F50-0x00000F58 OK	Motherboard resources	IRQ 85	Microsoft ACPI-Compliant System	OK	IRQ 116	Microsoft ACPI-Compliant System	OK
0x000000F0-0x000000F0 OK	Motherboard resources	IRQ 86	Microsoft ACPI-Compliant System	OK	IRQ 117	Microsoft ACPI-Compliant System	OK
0x00000CA0-0x00000CA1 OK	Motherboard resources	IRQ 87	Microsoft ACPI-Compliant System	OK	IRQ 118	Microsoft ACPI-Compliant System	OK
0x00000CA4-0x00000CA5 OK	Motherboard resources	IRQ 88	Microsoft ACPI-Compliant System	OK	IRQ 119	Microsoft ACPI-Compliant System	OK
0x000002F8-0x000002FF OK	Motherboard resources	IRQ 89	Microsoft ACPI-Compliant System	OK	IRQ 120	Microsoft ACPI-Compliant System	OK
0x00000CA2-0x00000CA3 Compliant Device OK	Microsoft Generic IPMI	IRQ 90	Microsoft ACPI-Compliant System	OK	IRQ 121	Microsoft ACPI-Compliant System	OK
0x00000040-0x00000043	System timer OK	IRQ 91	Microsoft ACPI-Compliant System	OK	IRQ 122	Microsoft ACPI-Compliant System	OK
0x00000080-0x0000008F controller OK	Direct memory access	IRQ 92	Microsoft ACPI-Compliant System	OK	IRQ 123	Microsoft ACPI-Compliant System	OK
0x000000C0-0x000000DF controller OK	Direct memory access	IRQ 93	Microsoft ACPI-Compliant System	OK	IRQ 124	Microsoft ACPI-Compliant System	OK
0x00000061-0x00000061	System speaker OK	IRQ 94	Microsoft ACPI-Compliant System	OK	IRQ 125	Microsoft ACPI-Compliant System	OK
0x00000060-0x00000060 OK	Standard PS/2 Keyboard	IRQ 95	Microsoft ACPI-Compliant System	OK	IRQ 126	Microsoft ACPI-Compliant System	OK
0x00000064-0x00000064 OK	Standard PS/2 Keyboard	IRQ 96	Microsoft ACPI-Compliant System	OK	IRQ 127	Microsoft ACPI-Compliant System	OK
0x0000002E-0x0000002F	Extended IO Bus OK	IRQ 97	Microsoft ACPI-Compliant System	OK	IRQ 128	Microsoft ACPI-Compliant System	OK
0x00000620-0x0000065F	Extended IO Bus OK	IRQ 98	Microsoft ACPI-Compliant System	OK	IRQ 129	Microsoft ACPI-Compliant System	OK
0x00000680-0x0000069F	Extended IO Bus OK	IRQ 99	Microsoft ACPI-Compliant System	OK	IRQ 130	Microsoft ACPI-Compliant System	OK
0x00000600-0x0000061F	Extended IO Bus OK	IRQ 100	Microsoft ACPI-Compliant System	OK	IRQ 131	Microsoft ACPI-Compliant System	OK
0x00000660-0x0000067F	Extended IO Bus OK	IRQ 101	Microsoft ACPI-Compliant System	OK	IRQ 132	Microsoft ACPI-Compliant System	OK
0x00000300-0x0000031F	Extended IO Bus OK	IRQ 102	Microsoft ACPI-Compliant System	OK	IRQ 133	Microsoft ACPI-Compliant System	OK
0x000003F8-0x000003FF (COM1) OK	Communications Port	IRQ 103	Microsoft ACPI-Compliant System	OK	IRQ 134	Microsoft ACPI-Compliant System	OK
0x00001080-0x00001087 PCI IDE Controller OK	Standard Dual Channel	IRQ 104	Microsoft ACPI-Compliant System	OK	IRQ 135	Microsoft ACPI-Compliant System	OK
0x00001088-0x0000108B PCI IDE Controller OK	Standard Dual Channel	IRQ 105	Microsoft ACPI-Compliant System	OK	IRQ 136	Microsoft ACPI-Compliant System	OK
0x00001090-0x00001097 PCI IDE Controller OK	Standard Dual Channel	IRQ 106	Microsoft ACPI-Compliant System	OK	IRQ 137	Microsoft ACPI-Compliant System	OK
0x00001098-0x0000109B PCI IDE Controller OK	Standard Dual Channel	IRQ 107	Microsoft ACPI-Compliant System	OK	IRQ 138	Microsoft ACPI-Compliant System	OK
0x000010A0-0x000010AF PCI IDE Controller OK	Standard Dual Channel	IRQ 108	Microsoft ACPI-Compliant System	OK	IRQ 139	Microsoft ACPI-Compliant System	OK
0x000010B0-0x000010BF PCI IDE Controller OK	Standard Dual Channel	IRQ 109	Microsoft ACPI-Compliant System	OK	IRQ 140	Microsoft ACPI-Compliant System	OK
[IRQs]		IRQ 110	Microsoft ACPI-Compliant System	OK	IRQ 141	Microsoft ACPI-Compliant System	OK
Resource Device Status		IRQ 111	Microsoft ACPI-Compliant System	OK	IRQ 142	Microsoft ACPI-Compliant System	OK

IRQ 143	Microsoft ACPI-Compliant System	OK	IRQ 174	Microsoft ACPI-Compliant System	OK	IRQ 4294967272	Smart Array P410i Controller	OK
IRQ 144	Microsoft ACPI-Compliant System	OK	IRQ 175	Microsoft ACPI-Compliant System	OK	IRQ 4294967271	Smart Array P410i Controller	OK
IRQ 145	Microsoft ACPI-Compliant System	OK	IRQ 176	Microsoft ACPI-Compliant System	OK	IRQ 4294967270	Smart Array P410i Controller	OK
IRQ 146	Microsoft ACPI-Compliant System	OK	IRQ 177	Microsoft ACPI-Compliant System	OK	IRQ 4294967269	Smart Array P410i Controller	OK
IRQ 147	Microsoft ACPI-Compliant System	OK	IRQ 178	Microsoft ACPI-Compliant System	OK	IRQ 4294967268	Smart Array P410i Controller	OK
IRQ 148	Microsoft ACPI-Compliant System	OK	IRQ 179	Microsoft ACPI-Compliant System	OK	IRQ 4294967267	Smart Array P410i Controller	OK
IRQ 149	Microsoft ACPI-Compliant System	OK	IRQ 180	Microsoft ACPI-Compliant System	OK	IRQ 4294967266	Smart Array P410i Controller	OK
IRQ 150	Microsoft ACPI-Compliant System	OK	IRQ 181	Microsoft ACPI-Compliant System	OK	IRQ 4294967265	Smart Array P410i Controller	OK
IRQ 151	Microsoft ACPI-Compliant System	OK	IRQ 182	Microsoft ACPI-Compliant System	OK	IRQ 4294967288	PCI standard PCI-to-PCI bridge	OK
IRQ 152	Microsoft ACPI-Compliant System	OK	IRQ 183	Microsoft ACPI-Compliant System	OK	IRQ 30	PCI standard PCI-to-PCI bridge	OK
IRQ 153	Microsoft ACPI-Compliant System	OK	IRQ 184	Microsoft ACPI-Compliant System	OK	IRQ 30	PCI standard PCI-to-PCI bridge	OK
IRQ 154	Microsoft ACPI-Compliant System	OK	IRQ 185	Microsoft ACPI-Compliant System	OK	IRQ 30	PCI standard PCI-to-PCI bridge	OK
IRQ 155	Microsoft ACPI-Compliant System	OK	IRQ 186	Microsoft ACPI-Compliant System	OK	IRQ 30	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 156	Microsoft ACPI-Compliant System	OK	IRQ 187	Microsoft ACPI-Compliant System	OK	IRQ 37	PCI standard PCI-to-PCI bridge	OK
IRQ 157	Microsoft ACPI-Compliant System	OK	IRQ 188	Microsoft ACPI-Compliant System	OK	IRQ 37	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 158	Microsoft ACPI-Compliant System	OK	IRQ 189	Microsoft ACPI-Compliant System	OK	IRQ 37	PCI standard PCI-to-PCI bridge	OK
IRQ 159	Microsoft ACPI-Compliant System	OK	IRQ 190	Microsoft ACPI-Compliant System	OK	IRQ 37	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 160	Microsoft ACPI-Compliant System	OK	IRQ 4294967294	PCI standard PCI-to-PCI bridge	OK	IRQ 37	PCI standard PCI-to-PCI bridge	OK
IRQ 161	Microsoft ACPI-Compliant System	OK	IRQ 28	Smart Array P411 Controller (Non-Miniport)	OK	IRQ 37	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 162	Microsoft ACPI-Compliant System	OK	IRQ 4294967282	PCI standard PCI-to-PCI bridge	OK	IRQ 39	PCI standard PCI-to-PCI bridge	OK
IRQ 163	Microsoft ACPI-Compliant System	OK	IRQ 4294967292	PCI standard PCI-to-PCI bridge	OK	IRQ 39	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 164	Microsoft ACPI-Compliant System	OK	IRQ 4294967284	PCI standard PCI-to-PCI bridge	OK	IRQ 39	PCI standard PCI-to-PCI bridge	OK
IRQ 165	Microsoft ACPI-Compliant System	OK	IRQ 4294967261	Intel(R) PRO/1000 PT Quad Port LP Server Adapter	OK	IRQ 39	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 166	Microsoft ACPI-Compliant System	OK	IRQ 4294967262	Intel(R) PRO/1000 PT Quad Port LP Server Adapter #2	OK	IRQ 4294967280	PCI standard PCI-to-PCI bridge	OK
IRQ 167	Microsoft ACPI-Compliant System	OK	IRQ 4294967283	PCI standard PCI-to-PCI bridge	OK	IRQ 4294967279	PCI standard PCI-to-PCI bridge	OK
IRQ 168	Microsoft ACPI-Compliant System	OK	IRQ 4294967263	Intel(R) PRO/1000 PT Quad Port LP Server Adapter #3	OK	IRQ 4294967278	PCI standard PCI-to-PCI bridge	OK
IRQ 169	Microsoft ACPI-Compliant System	OK	IRQ 4294967264	Intel(R) PRO/1000 PT Quad Port LP Server Adapter #4	OK	IRQ 16	PCI standard PCI-to-PCI bridge	OK
IRQ 170	Microsoft ACPI-Compliant System	OK	IRQ 4294967281	PCI standard PCI-to-PCI bridge	OK	IRQ 16	Smart Array P411 Controller (Non-Miniport)	OK
IRQ 171	Microsoft ACPI-Compliant System	OK	IRQ 4294967290	PCI standard PCI-to-PCI bridge	OK	IRQ 20	Standard Universal PCI to USB Host Controller	OK
IRQ 172	Microsoft ACPI-Compliant System	OK	IRQ 26	Smart Array P411 Controller (Non-Miniport)	OK	IRQ 20	Standard Enhanced PCI to USB Host Controller	OK
IRQ 173	Microsoft ACPI-Compliant System	OK	IRQ 4294967289	PCI standard PCI-to-PCI bridge	OK	IRQ 23	Standard Universal PCI to USB Host Controller	OK

```

IRQ 23 Standard Universal PCI to USB Host
Controller OK
IRQ 22 Standard Universal PCI to USB Host
Controller OK
IRQ 22 Standard Universal PCI to USB Host
Controller OK
IRQ 10 Base System Device OK
IRQ 10 Base System Device OK
IRQ 10 IPMI Interface OK
IRQ 0 System timer OK
IRQ 1 Standard PS/2 Keyboard OK
IRQ 12 PS/2 Compatible Mouse OK
IRQ 4 Communications Port (COM1) OK
IRQ 17 Standard Dual Channel PCI IDE Controller
OK
IRQ 4294967285 PCI standard PCI-to-PCI bridge
OK
IRQ 52 Smart Array P411 Controller (Non-Miniport)
OK
IRQ 4294967277 PCI standard PCI-to-PCI bridge
OK
IRQ 4294967286 PCI standard PCI-to-PCI bridge
OK
IRQ 48 Smart Array P411 Controller (Non-Miniport)
OK
IRQ 4294967276 PCI standard PCI-to-PCI bridge
OK
IRQ 4294967275 PCI standard PCI-to-PCI bridge
OK
IRQ 4294967274 PCI standard PCI-to-PCI bridge
OK
IRQ 4294967287 PCI standard PCI-to-PCI bridge
OK
IRQ 54 QLogic Fibre Channel Adapter OK
IRQ 61 QLogic Fibre Channel Adapter OK
IRQ 4294967291 PCI standard PCI-to-PCI bridge
OK
IRQ 55 Smart Array P411 Controller (Non-Miniport)
OK
IRQ 4294967293 PCI standard PCI-to-PCI bridge
OK
IRQ 56 Smart Array P411 Controller (Non-Miniport)
OK
IRQ 4294967273 PCI standard PCI-to-PCI bridge
OK

[Memory]

Resource Device Status
0xE7000000-0xF8FFFFFF PCI bus OK
0xFED00000-0xFED03FFF PCI bus OK
0xFED00000-0xFED03FFF PCI bus OK
0xFED00000-0xFED03FFF High precision event
timer OK
0xA0000-0xBFFFF PCI bus OK
0xA0000-0xBFFFF Standard VGA Graphics Adapter OK

0xF5300000-0xF57FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5400000-0xF57FFFFFF Smart Array P411
Controller (Non-Miniport) OK

```

```

0xF53F0000-0xF53F0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF5100000-0xF52FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5100000-0xF52FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5100000-0xF52FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF51E0000-0xF51FFFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter OK
0xF51C0000-0xF51DFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter OK
0xF51A0000-0xF51BFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #2 OK
0xF5180000-0xF519FFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #2 OK
0xF5200000-0xF52FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF52E0000-0xF52FFFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #3 OK
0xF52C0000-0xF52DFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #3 OK
0xF52A0000-0xF52BFFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #4 OK
0xF5280000-0xF529FFFF Intel(R) PRO/1000 PT
Quad Port LP Server Adapter #4 OK
0xF8B00000-0xF8FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF8C00000-0xF8FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF8BF0000-0xF8BF0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF4B00000-0xF4FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF4C00000-0xF4FFFFFF Smart Array P410i
Controller OK
0xF4BF0000-0xF4BF0FFF Smart Array P410i
Controller OK
0xF5A00000-0xF87FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5AE0000-0xF5AFFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5B00000-0xF87FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5B00000-0xF87FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF5C00000-0xF5FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF5BF0000-0xF5BF0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF6300000-0xF67FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF6400000-0xF67FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF63F0000-0xF63F0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF6B00000-0xF6FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF6C00000-0xF6FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF6BF0000-0xF6BF0FFF Smart Array P411
Controller (Non-Miniport) OK

```

```

0xF7300000-0xF77FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF7400000-0xF77FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF73F0000-0xF73F0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF7B00000-0xF7FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF7C00000-0xF7FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF7BF0000-0xF7BF0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF8300000-0xF87FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF8400000-0xF87FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF83F0000-0xF83F0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF4300000-0xF47FFFFFF PCI standard PCI-to-PCI
bridge OK
0xF4400000-0xF47FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF43F0000-0xF43F0FFF Smart Array P411
Controller (Non-Miniport) OK
0xF40F0000-0xF40F3FFF Standard Enhanced PCI
to USB Host Controller OK
0xF4100000-0xF42FFFFFF Intel(R) 82801 PCI
Bridge - 244E OK
0xE8000000-0xEFFFFFFF Intel(R) 82801 PCI
Bridge - 244E OK
0xE8000000-0xEFFFFFFF Standard VGA Graphics
Adapter OK
0xF42F0000-0xF42FFFFFF Standard VGA Graphics
Adapter OK
0xF42E0000-0xF42E01FF Base System Device OK
0xF42D0000-0xF42D07FF Base System Device OK
0xF42C0000-0xF42C3FFF Base System Device OK
0xF4200000-0xF427FFFF Base System Device OK
0xF41F0000-0xF41F00FF IPMI Interface OK
0xE0000000-0xE3FFFFFF Motherboard resources
OK
0xFE000000-0xFEBFFFFFF Motherboard resources
OK
0xF9000000-0xFBFFFFFF PCI bus OK
0xF9F00000-0xFA3FFFFFF PCI standard PCI-to-PCI
bridge OK
0xFA000000-0xFA3FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xF9FF0000-0xF9FF0FFF Smart Array P411
Controller (Non-Miniport) OK
0xFB700000-0xFB7FFFFFF PCI standard PCI-to-PCI
bridge OK
0xFB800000-0xFB7FFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xFB7F0000-0xFB7F0FFF Smart Array P411
Controller (Non-Miniport) OK

```

```

0xFBFB0000-0xFBFBFFFF PCI standard PCI-to-PCI
bridge OK
0xFBFB0000-0xFBFB3FFF QLogic Fibre Channel
Adapter OK
0xFBFB0000-0xFBFB3FFF QLogic Fibre Channel
Adapter OK
0xFAF00000-0xFB3FFFFF PCI standard PCI-to-PCI
bridge OK
0xFB000000-0xFB3FFFFF Smart Array P411
Controller (Non-Miniport) OK
0xFAF00000-0xFAFF0FFF Smart Array P411
Controller (Non-Miniport) OK
0xFA700000-0xFABFFFFF PCI standard PCI-to-PCI
bridge OK
0xFA800000-0xFABFFFFF Smart Array P411
Controller (Non-Miniport) OK
0xFA7F0000-0xFA7F0FFF Smart Array P411
Controller (Non-Miniport) OK

```

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
c:\windows\system32\msadp32.acm	Microsoft Corporation	MSADP32.ACM	OK	C:\Windows\system32\MSADP32.ACM	6.0.6000.16386	22.00 KB (22,528 bytes)
c:\windows\system32\msg711.acm	Microsoft Corporation	MSG711.ACM	OK	C:\Windows\system32\MSG711.ACM	6.0.6000.16386	14.00 KB (14,336 bytes)
c:\windows\system32\msgsm32.acm	Microsoft Corporation	MSGSM32.ACM	OK	C:\Windows\system32\MSGSM32.ACM	6.0.6000.16386	28.00 KB (28,672 bytes)
c:\windows\system32\imaadp32.acm	Microsoft Corporation	IMAADP32.ACM	OK	C:\Windows\system32\IMAADP32.ACM	6.0.6000.16386	21.00 KB (21,504 bytes)
c:\windows\system32\tsssoft32.acm	DSP GROUP, INC.	TSSOFT32.ACM	OK	C:\Windows\system32\tsssoft32.acm	1.1.1.5	13.50 KB (13,824 bytes)

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
c:\windows\system32\iyuv_32.dll	Microsoft Corporation	IYUV_32.DLL	OK	C:\Windows\system32\IYUV_32.DLL	6.0.6000.16386	52.50 KB (53,760 bytes)
c:\windows\system32\msrle32.dll	Microsoft Corporation	MSRLE32.DLL	OK	C:\Windows\system32\MSRLE32.DLL	6.0.6000.16386	15.50 KB (15,872 bytes)
c:\windows\system32\msvidc32.dll	Microsoft Corporation	MSVIDC32.DLL	OK	C:\Windows\system32\MSVIDC32.DLL	6.0.6001.18000	37.50 KB (38,400 bytes)
c:\windows\system32\msyuv.dll	Microsoft Corporation	MSYUV.DLL	OK	C:\Windows\system32\MSYUV.DLL	6.0.6000.16386	24.50 KB (25,088 bytes)
c:\windows\system32\tsbyuv.dll	Microsoft Corporation	TSBYUV.DLL	OK	C:\Windows\system32\TSBYUV.DLL	6.0.6002.16670	14.00 KB (14,336 bytes)

[CD-ROM]

Item	Value
Drive D:	CD-ROM Drive
Media Loaded	No
Media Type	UNKNOWN
Name	HL-DT-ST DVD-ROM GDRH20N ATA Device
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	-1.00 kbytes/sec
SCSI Target ID	0
PNP Device ID	IDE\CDROMHL-DT-ST_DVD-ROM_GDRH20N_08E4_5&2C0B779D&0&1.0.0
Driver	c:\windows\system32\drivers\cdrom.sys (6.0.6002.16670, 78.00 KB (79,872 bytes), 10/28/2009 10:31 AM)

[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 Type	Not Available, (Standard display types) compatible
Adapter Description	Standard VGA Graphics Adapter

Item	Value
Adapter RAM	Not Available
Installed Drivers	Not Available
Driver Version	6.0.6001.18000
INF File	display.inf (vga section)
Color Planes	Not Available
Color Table Entries	Not Available
Resolution	Not Available
Bits/Pixel	Not Available
Memory Address	0xE8000000-0xEFFFFFFF
I/O Port	0x00003000-0x000030FF
Memory Address	0xF42F0000-0xF42FFFFF
I/O Port	0x000003B0-0x000003BB
I/O Port	0x000003C0-0x000003DF
Memory Address	0xA0000-0xBFFFFF
Driver	c:\windows\system32\drivers\vgapnp.sys (6.0.6001.18000, 28.50 KB (29,184 bytes), 1/19/2008 3:38 AM)

[Infrared]

Item	Value
------	-------

[Input]

[Keyboard]

Item	Value
Description	USB Human Interface 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\hidusb.sys (6.0.6002.16670, 15.50 KB (15,872 bytes), 10/28/2009 10:31 AM)
Description	Standard PS/2 Keyboard
Name	Enhanced (101- or 102-key)
Layout	00000409
PNP Device ID	ACPI\PNP0303\4&1CFF2C4C&0
Number of Function Keys	12
I/O Port	0x00000060-0x00000060
I/O Port	0x00000064-0x00000064
IRQ Channel	IRQ 1
Driver	c:\windows\system32\drivers\i8042prt.sys (6.0.6001.18000, 62.50 KB (64,000 bytes), 1/19/2008 12:28 AM)

[Pointing Device]

Item	Value
Hardware Type	USB Human Interface 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\hidusb.sys
(6.0.6002.16670, 15.50 KB (15,872 bytes), 10/28/2009
10:31 AM)

Hardware Type PS/2 Compatible Mouse
Number of Buttons 0
Status OK
PNP Device ID ACPI\PNP0F13\4&1CFF2C4C&0
Power Management Supported No
Double Click Threshold Not Available
Handedness Not Available
IRQ Channel IRQ 12
Driver c:\windows\system32\drivers\i8042prt.sys
(6.0.6001.18000, 62.50 KB (64,000 bytes), 1/19/2008
12:28 AM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000000] WAN Miniport (SSTP)
Adapter Type Not Available
Product Type WAN Miniport (SSTP)
Installed Yes
PNP Device ID ROOT\MS_SSTP\Miniport\0000
Last Reset 11/3/2009 8:31 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\rassstp.sys
(6.0.6002.16670, 76.50 KB (78,336 bytes), 10/28/2009
10:31 AM)

Name [00000001] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TP\Miniport\0000
Last Reset 11/3/2009 8:31 AM
Index 1
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
(6.0.6002.16670, 122.50 KB (125,440 bytes),
10/28/2009 10:31 AM)

Name [00000002] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTP\Miniport\0000
Last Reset 11/3/2009 8:31 AM
Index 2
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys
(6.0.6002.16670, 96.50 KB (98,816 bytes), 10/28/2009
10:31 AM)

Name [00000003] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOE\Miniport\0000
Last Reset 11/3/2009 8:31 AM
Index 3
Service Name Rasppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\rasppoe.sys
(6.0.6002.16670, 49.00 KB (50,176 bytes), 10/28/2009
10:31 AM)

Name [00000004] WAN Miniport (IPv6)
Adapter Type Not Available
Product Type WAN Miniport (IPv6)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIPV6\0000
Last Reset 11/3/2009 8:31 AM
Index 4
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
(6.0.6002.16670, 165.00 KB (168,960 bytes),
10/28/2009 10:31 AM)

Name [00000005] WAN Miniport (Network Monitor)

Adapter Type Not Available
Product Type WAN Miniport (Network Monitor)

Installed Yes
PNP Device ID ROOT\MS_NDISWANBH\0000
Last Reset 11/3/2009 8:31 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
(6.0.6002.16670, 165.00 KB (168,960 bytes),
10/28/2009 10:31 AM)

Name [00000006] HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Adapter Type Not Available
Product Type HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 8:31 AM
Index 6
Service Name NXND6HP
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 [00000007] HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Adapter Type Not Available
Product Type HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 8:31 AM
Index 7
Service Name NXND6HP
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 [00000008] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 11/3/2009 8:31 AM
Index 8
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys
(6.0.6002.16670, 165.00 KB (168,960 bytes),
10/28/2009 10:31 AM)

Name [00000009] Intel(R) PRO/1000 PT Quad Port
LP Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Quad Port LP
Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&21E81E70&0&00000018
Last Reset 11/3/2009 8:31 AM
Index 9
Service Name elexpress
IP Address 130.168.208.20
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:18:FE:2E:1E:BD
Memory Address 0xF51E0000-0xF51FFFFF
Memory Address 0xF51C0000-0xF51DFFFF
IRQ Channel IRQ 4294967261
Driver c:\windows\system32\drivers\ele6032e.sys
(9.11.5.7, 310.50 KB (317,952 bytes), 1/19/2008 3:38
AM)

Name [00000010] Intel(R) PRO/1000 PT Quad Port
LP Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Quad Port LP
Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&21E81E70&0&01000018
Last Reset 11/3/2009 8:31 AM
Index 10
Service Name elexpress

IP Address 130.130.208.1
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:18:FE:2E:1E:BC
Memory Address 0xF51A0000-0xF51BFFFF
Memory Address 0xF5180000-0xF519FFFF
IRQ Channel IRQ 4294967262
Driver c:\windows\system32\drivers\ele6032e.sys
(9.11.5.7, 310.50 KB (317,952 bytes), 1/19/2008 3:38
AM)

Name [00000011] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 8:31 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 Not Available

Name [00000012] HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Adapter Type Not Available
Product Type HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 8:31 AM
Index 12
Service Name NXND6HP
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 [00000013] HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Adapter Type Not Available
Product Type HP NC375i Integrated Quad Port
Multifunction Gigabit Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 8:31 AM
Index 13
Service Name NXND6HP
IP Address Not Available

IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000014] Microsoft ISATAP Adapter
Adapter Type Tunnel
Product Type Microsoft ISATAP Adapter
Installed Yes
PNP Device ID ROOT*ISATAP\0000
Last Reset 11/3/2009 8:31 AM
Index 14
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
(6.0.6001.18000, 27.50 KB (28,160 bytes), 1/19/2008
12:36 AM)

Name [00000015] Intel(R) PRO/1000 PT Quad Port
LP Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Quad Port LP
Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&4CC682A&0&00080018
Last Reset 11/3/2009 8:31 AM
Index 15
Service Name elexpress
IP Address 130.131.208.2
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 00:18:FE:2E:1E:BF
Memory Address 0xF52E0000-0xF52FFFFF
Memory Address 0xF52C0000-0xF52DFFFF
IRQ Channel IRQ 4294967263
Driver c:\windows\system32\drivers\ele6032e.sys
(9.11.5.7, 310.50 KB (317,952 bytes), 1/19/2008 3:38
AM)

Name [00000016] Intel(R) PRO/1000 PT Quad Port
LP Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Quad Port LP
Server Adapter
Installed Yes

```
PNP Device ID
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&4CC682A&0&01080018
Last Reset 11/3/2009 8:31 AM
Index 16
Service Name elexpress
IP Address 130.132.208.3
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 00:18:FE:2E:1E:BE
Memory Address 0xF52A0000-0xF52BFFFF
Memory Address 0xF5280000-0xF529FFFF
IRQ Channel IRQ 4294967264
Driver c:\windows\system32\drivers\ele6032e.sys
(9.11.5.7, 310.50 KB (317,952 bytes), 1/19/2008 3:38
AM)
```

```
Name [0000017] Microsoft 6to4 Adapter
Adapter Type Tunnel
Product Type Microsoft 6to4 Adapter
Installed Yes
PNP Device ID ROOT\*6TO4MP\0001
Last Reset 11/3/2009 8:31 AM
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\tunnel.sys
(6.0.6001.18000, 27.50 KB (28,160 bytes), 1/19/2008
12:36 AM)
```

```
Name [0000021] Microsoft ISATAP Adapter
Adapter Type Tunnel
Product Type Microsoft ISATAP Adapter
Installed Yes
PNP Device ID ROOT\*ISATAP\0004
Last Reset 11/3/2009 8:31 AM
Index 21
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
(6.0.6001.18000, 27.50 KB (28,160 bytes), 1/19/2008
12:36 AM)
```

```
Name [0000022] Microsoft ISATAP Adapter
```

```
Adapter Type Tunnel
Product Type Microsoft ISATAP Adapter
Installed Yes
PNP Device ID ROOT\*ISATAP\0005
Last Reset 11/3/2009 8:31 AM
Index 22
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
(6.0.6001.18000, 27.50 KB (28,160 bytes), 1/19/2008
12:36 AM)
```

```
Name [0000023] Microsoft ISATAP Adapter
Adapter Type Tunnel
Product Type Microsoft ISATAP Adapter
Installed Yes
PNP Device ID ROOT\*ISATAP\0006
Last Reset 11/3/2009 8:31 AM
Index 23
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
(6.0.6001.18000, 27.50 KB (28,160 bytes), 1/19/2008
12:36 AM)
```

```
[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
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\wsock32.dll
Size      15.00 KB (15,360 bytes)
Version   6.0.6001.18000

File      c:\windows\system32\wsock32.dll
Size      18.00 KB (18,432 bytes)
Version   6.0.6001.18000

[Ports]

[Serial]

Item      Value
Name      Communications Port (COM1)
Status    OK
PNP Device ID      ACPI\PNP0501\0
Maximum Input Buffer Size      0
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 4
I/O Port        0x000003F8-0x000003FF
Driver           c:\windows\system32\drivers\serial.sys
(6.0.6001.18000, 92.00 KB (94,208 bytes), 1/19/2008
12:28 AM)

[Parallel]

Item      Value

[Storage]

[Drives]

Item      Value
Drive     C:
Description      Local Fixed Disk
Compressed       No
File System      NTFS
Size             33.88 GB (36,381,310,976 bytes)
Free Space       5.57 GB (5,984,473,088 bytes)
Volume Name
Volume Serial Number      F863586B

Drive     D:
Description      CD-ROM Disc

Drive     E:
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     F:
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     N:
Description      Local Fixed Disk
Compressed       No
File System      NTFS
Size             1.45 TB (1,594,906,431,488 bytes)
Free Space       1.15 TB (1,263,101,050,880 bytes)

Volume Name      back11
Volume Serial Number      98A677AC

Drive     O:
Description      Local Fixed Disk
Compressed       No
File System      NTFS
Size             1.45 TB (1,594,906,431,488 bytes)

```



```

Free Space          1.15 TB (1,263,101,186,048 bytes)

Volume Name        back12
Volume Serial Number 04BD8BA3

Drive P:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,092,142,080 bytes)

Volume Name        back13
Volume Serial Number 04C837E5

Drive Q:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,101,046,784 bytes)

Volume Name        back6
Volume Serial Number DC5B0C8E

Drive R:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,101,054,976 bytes)

Volume Name        back7
Volume Serial Number 386B2E15

Drive S:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,101,046,784 bytes)

Volume Name        back8
Volume Serial Number 047A1957

Drive T:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,101,140,992 bytes)

Volume Name        back9
Volume Serial Number 688628FC

Drive U:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,101,140,992 bytes)

```

```

Volume Name        back10
Volume Serial Number 5490D9EF

Drive V:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,100,985,344 bytes)

Volume Name        back1
Volume Serial Number F01B13AC

Drive W:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.14 TB (1,258,416,910,336 bytes)

Volume Name        back2
Volume Serial Number E828F412

Drive X:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.01 TB (1,108,179,443,712 bytes)

Volume Name        back3
Volume Serial Number D833D719

Drive Y:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.01 TB (1,108,493,123,584 bytes)

Volume Name        back4
Volume Serial Number 4041BC32

Drive Z:
Description        Local Fixed Disk
Compressed         No
File System        NTFS
Size               1.45 TB (1,594,906,431,488 bytes)
Free Space         1.15 TB (1,263,084,478,464 bytes)

Volume Name        back5
Volume Serial Number 7C4FEDFD

[Disks]

Item Value
Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk

```

```

Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 166.01 GB (178,250,042,880 bytes)
Total Cylinders 21,671
Total Sectors 348,144,615
Total Tracks 5,526,105
Tracks/Cylinder 255
Partition Disk #45, Partition #0
Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 104.49 GB (112,192,819,200 bytes)
Total Cylinders 13,640
Total Sectors 219,126,600
Total Tracks 3,478,200
Tracks/Cylinder 255
Partition Disk #46, Partition #0
Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 118.16 GB (126,874,944,000 bytes)
Total Cylinders 15,425
Total Sectors 247,802,625
Total Tracks 3,933,375
Tracks/Cylinder 255
Partition Disk #47, Partition #0
Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard

```

Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #48, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

 Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #49, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

 Partition Starting Offset 32,256 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #5, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

 Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

 Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120

Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

 Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

 Partition Starting Offset 32,256 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #60, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

 Partition Starting Offset 131,072 bytes

 Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63

Size 104.49 GB (112,192,819,200 bytes)
Total Cylinders 13,640
Total Sectors 219,126,600
Total Tracks 3,478,200
Tracks/Cylinder 255
Partition Disk #61, Partition #0
Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 118.16 GB (126,874,944,000 bytes)
Total Cylinders 15,425
Total Sectors 247,802,625
Total Tracks 3,933,375
Tracks/Cylinder 255
Partition Disk #62, Partition #0
Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 29.29 GB (31,453,470,720 bytes)
Total Cylinders 3,824
Total Sectors 61,432,560
Total Tracks 975,120
Tracks/Cylinder 255
Partition Disk #63, Partition #0
Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available

SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 1.45 TB (1,594,906,467,840 bytes)
Total Cylinders 193,903
Total Sectors 3,115,051,695
Total Tracks 49,445,265
Tracks/Cylinder 255
Partition Disk #64, Partition #0
Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 166.01 GB (178,250,042,880 bytes)
Total Cylinders 21,671
Total Sectors 348,144,615
Total Tracks 5,526,105
Tracks/Cylinder 255
Partition Disk #55, Partition #0
Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 104.49 GB (112,192,819,200 bytes)
Total Cylinders 13,640
Total Sectors 219,126,600
Total Tracks 3,478,200
Tracks/Cylinder 255
Partition Disk #56, Partition #0
Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512

Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 118.16 GB (126,874,944,000 bytes)
Total Cylinders 15,425
Total Sectors 247,802,625
Total Tracks 3,933,375
Tracks/Cylinder 255
Partition Disk #57, Partition #0
Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 29.29 GB (31,453,470,720 bytes)
Total Cylinders 3,824
Total Sectors 61,432,560
Total Tracks 975,120
Tracks/Cylinder 255
Partition Disk #58, Partition #0
Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 1.45 TB (1,594,906,467,840 bytes)
Total Cylinders 193,903
Total Sectors 3,115,051,695
Total Tracks 49,445,265
Tracks/Cylinder 255
Partition Disk #59, Partition #0
Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #0, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #1, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0

Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #4, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671

Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #40, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #41, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #42, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available

SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #43, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #44, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #50, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #51, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #52, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #53, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard

Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #54, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #35, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #36, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #37, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #38, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265

Tracks/Cylinder 255
 Partition Disk #39, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #10, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63

Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #12, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #13, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #14, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available

SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 166.01 GB (178,250,042,880 bytes)
Total Cylinders 21,671
Total Sectors 348,144,615
Total Tracks 5,526,105
Tracks/Cylinder 255
Partition Disk #25, Partition #0
Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 104.49 GB (112,192,819,200 bytes)
Total Cylinders 13,640
Total Sectors 219,126,600
Total Tracks 3,478,200
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 118.16 GB (126,874,944,000 bytes)
Total Cylinders 15,425
Total Sectors 247,802,625
Total Tracks 3,933,375
Tracks/Cylinder 255
Partition Disk #27, Partition #0
Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512

Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 29.29 GB (31,453,470,720 bytes)
Total Cylinders 3,824
Total Sectors 61,432,560
Total Tracks 975,120
Tracks/Cylinder 255
Partition Disk #28, Partition #0
Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 1.45 TB (1,594,906,467,840 bytes)
Total Cylinders 193,903
Total Sectors 3,115,051,695
Total Tracks 49,445,265
Tracks/Cylinder 255
Partition Disk #29, Partition #0
Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 166.01 GB (178,250,042,880 bytes)
Total Cylinders 21,671
Total Sectors 348,144,615
Total Tracks 5,526,105
Tracks/Cylinder 255
Partition Disk #30, Partition #0
Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 104.49 GB (112,192,819,200 bytes)
Total Cylinders 13,640
Total Sectors 219,126,600
Total Tracks 3,478,200
Tracks/Cylinder 255
Partition Disk #31, Partition #0
Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 118.16 GB (126,874,944,000 bytes)
Total Cylinders 15,425
Total Sectors 247,802,625
Total Tracks 3,933,375
Tracks/Cylinder 255
Partition Disk #32, Partition #0
Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
Manufacturer Hewlett-Packard
Model HP LOGICAL VOLUME Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus Not Available
SCSI Logical Unit Not Available
SCSI Port Not Available
SCSI Target ID Not Available
Sectors/Track 63
Size 29.29 GB (31,453,470,720 bytes)
Total Cylinders 3,824
Total Sectors 61,432,560
Total Tracks 975,120
Tracks/Cylinder 255
Partition Disk #33, Partition #0

Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #34, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640

Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #22, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #23, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available

SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 166.01 GB (178,250,042,880 bytes)
 Total Cylinders 21,671
 Total Sectors 348,144,615
 Total Tracks 5,526,105
 Tracks/Cylinder 255
 Partition Disk #15, Partition #0
 Partition Size 166.01 GB (178,249,531,392 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 104.49 GB (112,192,819,200 bytes)
 Total Cylinders 13,640
 Total Sectors 219,126,600
 Total Tracks 3,478,200
 Tracks/Cylinder 255
 Partition Disk #16, Partition #0
 Partition Size 104.49 GB (112,192,389,120 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk

Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 118.16 GB (126,874,944,000 bytes)
 Total Cylinders 15,425
 Total Sectors 247,802,625
 Total Tracks 3,933,375
 Tracks/Cylinder 255
 Partition Disk #17, Partition #0
 Partition Size 118.16 GB (126,874,550,272 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 29.29 GB (31,453,470,720 bytes)
 Total Cylinders 3,824
 Total Sectors 61,432,560
 Total Tracks 975,120
 Tracks/Cylinder 255
 Partition Disk #18, Partition #0
 Partition Size 29.29 GB (31,453,085,696 bytes)

Partition Starting Offset 131,072 bytes

Description Smart Array Logical Volume
 Manufacturer Hewlett-Packard
 Model HP LOGICAL VOLUME Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 1
 SCSI Bus Not Available
 SCSI Logical Unit Not Available
 SCSI Port Not Available
 SCSI Target ID Not Available
 Sectors/Track 63
 Size 1.45 TB (1,594,906,467,840 bytes)
 Total Cylinders 193,903
 Total Sectors 3,115,051,695
 Total Tracks 49,445,265
 Tracks/Cylinder 255
 Partition Disk #19, Partition #0
 Partition Size 1.45 TB (1,594,906,435,584 bytes)

Partition Starting Offset 32,256 bytes

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 1
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 2
 SCSI Target ID 4
 Sectors/Track 32
 Size 33.89 GB (36,385,505,280 bytes)
 Total Cylinders 8,709
 Total Sectors 71,065,440
 Total Tracks 2,220,795
 Tracks/Cylinder 255
 Partition Disk #65, Partition #0
 Partition Size 33.88 GB (36,381,310,976 bytes)

Partition Starting Offset 16,384 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 4
 SCSI Target ID 0
 Sectors/Track 63
 Size 1.07 TB (1,172,998,955,520 bytes)
 Total Cylinders 142,609
 Total Sectors 2,291,013,585
 Total Tracks 36,365,295
 Tracks/Cylinder 255
 Partition Disk #67, Partition #0
 Partition Size 1.07 TB (1,172,998,914,560 bytes)

Partition Starting Offset 32,256 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 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 1.07 TB (1,172,998,955,520 bytes)
 Total Cylinders 142,609
 Total Sectors 2,291,013,585
 Total Tracks 36,365,295
 Tracks/Cylinder 255
 Partition Disk #66, Partition #0
 Partition Size 1.07 TB (1,172,998,914,560 bytes)

Partition Starting Offset 32,256 bytes

[SCSI]

Item Value
 Name Smart Array P411 Controller (Non-Miniport)

Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&38B514C2&0&0008
 Memory Address 0xF5400000-0xF57FFFFF
 Memory Address 0xF53F0000-0xF53F0FFF
 IRQ Channel IRQ 28
 Driver c:\windows\system32\drivers\hpcqissb.sys (6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)

Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&18716765&0&0028
 Memory Address 0xF8C00000-0xF8FFFFFF
 Memory Address 0xF8BF0000-0xF8BF0FFF
 IRQ Channel IRQ 26
 Driver c:\windows\system32\drivers\hpcqissb.sys (6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 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&2B9E441F&0&0030

Memory Address 0xF4C00000-0xF4FFFFFF
 Memory Address 0xF4BF0000-0xF4BF0FFF
 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
 Driver c:\windows\system32\drivers\hpciss2.sys (6.20.0.64, 147.04 KB (150,568 bytes), 10/28/2009 8:34 AM)

Name Smart Array P411 Controller (Non-Miniport)

Manufacturer Hewlett-Packard
 Status OK
 PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&1626DA6&0&00080038
 Memory Address 0xF5C00000-0xF5FFFFFF
 Memory Address 0xF5BF0000-0xF5BF0FFF

IRQ Channel IRQ 37
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&3B2C24D1&0&00480038
Memory Address 0xF6400000-0xF67FFFFF
Memory Address 0xF63F0000-0xF63F0FFF
IRQ Channel IRQ 37
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&179CC572&0&00500038
Memory Address 0xF6C00000-0xF6FFFFFF
Memory Address 0xF6BF0000-0xF6BF0FFF
IRQ Channel IRQ 39
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&326DB53E&0&00600038
Memory Address 0xF7400000-0xF77FFFFF
Memory Address 0xF73F0000-0xF73F0FFF
IRQ Channel IRQ 30
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&2FE19B63&0&00680038
Memory Address 0xF7C00000-0xF7FFFFFF
Memory Address 0xF7BF0000-0xF7BF0FFF
IRQ Channel IRQ 37
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&C523C04&0&00700038
Memory Address 0xF8400000-0xF87FFFFF
Memory Address 0xF83F0000-0xF83F0FFF
IRQ Channel IRQ 39
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&294C1E27&0&00E0
Memory Address 0xF4400000-0xF47FFFFF
Memory Address 0xF43F0000-0xF43F0FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&3892F0CE&0&0008
Memory Address 0xFA000000-0xFA3FFFFFFF
Memory Address 0xF9FF0000-0xF9FF0FFF
IRQ Channel IRQ 52
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&59E450C&0&0018
Memory Address 0xFB800000-0xFBFFFFFFF
Memory Address 0xFB7F0000-0xFB7F0FFF
IRQ Channel IRQ 48
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&2B0041D6&0&0038

Memory Address 0xFBFF0000-0xFBFF3FFF
IRQ Channel IRQ 54
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.8.17, 1.11 MB (1,160,232 bytes), 10/29/2009 10:53 AM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_02\4&2B0041D6&0&0138
Memory Address 0xFBFE0000-0xFBFE3FFF
IRQ Channel IRQ 61
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.8.17, 1.11 MB (1,160,232 bytes), 10/29/2009 10:53 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&190F69AE&0&0040
Memory Address 0xFB000000-0xFB3FFFFFFF
Memory Address 0xFAFF0000-0xFAFF0FFF
IRQ Channel IRQ 55
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Smart Array P411 Controller (Non-Miniport)
Manufacturer Hewlett-Packard
Status OK
PNP Device ID PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&11FF4A0C&0&0048
Memory Address 0xFA800000-0xFABFFFFFFF
Memory Address 0xFA7F0000-0xFA7F0FFF
IRQ Channel IRQ 56
Driver c:\windows\system32\drivers\hpcqissb.sys
(6.4.0.64, 66.54 KB (68,136 bytes), 1/14/2009 11:13 AM)

Name Microsoft iSCSI Initiator
Manufacturer Microsoft
Status OK
PNP Device ID ROOT\ISCSIPRT\0000
Driver c:\windows\system32\drivers\msiscsi.sys
(6.0.6002.16670, 209.48 KB (214,504 bytes), 10/28/2009 10:31 AM)

[IDE]

Item Value
Name Standard Dual Channel PCI IDE Controller
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK

```

PNP Device ID
    PCI\VEN_8086&DEV_3A20&SUBSYS_330D103C&REV_0
0\3&267A616A&0&FA
I/O Port 0x00001080-0x00001087
I/O Port 0x00001088-0x0000108B
I/O Port 0x00001090-0x00001097
I/O Port 0x00001098-0x0000109B
I/O Port 0x000010A0-0x000010AF
I/O Port 0x000010B0-0x000010BF
IRQ Channel  IRQ 17
Driver c:\windows\system32\drivers\pciide.sys
(6.0.6002.16670, 13.98 KB (14,312 bytes), 10/28/2009
10:31 AM)

Name IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&CF23815&0&0

Driver c:\windows\system32\drivers\atapi.sys
(6.0.6002.16670, 20.46 KB (20,952 bytes), 10/28/2009
10:31 AM)

Name IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&CF23815&0&1

Driver c:\windows\system32\drivers\atapi.sys
(6.0.6002.16670, 20.46 KB (20,952 bytes), 10/28/2009
10:31 AM)

[Printing]

Can't Collect Information

[Problem Devices]

Device PNP Device ID Error Code
Base System Device
    PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_0
3\4&10AF73B4&0&20F0 This device is not configured
correctly.
Base System Device
    PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_0
3\4&10AF73B4&0&22F0 This device is not configured
correctly.
IPMI Interface
    PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_0
0\4&10AF73B4&0&26F0 This device is not configured
correctly.

[USB]

Device PNP Device ID
Standard Universal PCI to USB Host Controller
    PCI\VEN_8086&DEV_3A34&SUBSYS_330D103C&REV_0
0\3&267A616A&0&E8

```

```

Standard Universal PCI to USB Host Controller
    PCI\VEN_8086&DEV_3A35&SUBSYS_330D103C&REV_0
0\3&267A616A&0&E9
Standard Universal PCI to USB Host Controller
    PCI\VEN_8086&DEV_3A36&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EA
Standard Universal PCI to USB Host Controller
    PCI\VEN_8086&DEV_3A39&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EB
Standard Enhanced PCI to USB Host Controller
    PCI\VEN_8086&DEV_3A3A&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EF
Standard Universal PCI to USB Host Controller
    PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_0
0\4&10AF73B4&0&24F0

[Software Environment]

[System Drivers]

Name Description File Type
Started Start Mode State
Status Error Control Accept Pause
Accept Stop
acpi Microsoft ACPI Driver
    c:\windows\system32\drivers\acpi.sys
Kernel Driver Yes Boot
Running OK Critical No Yes

adp94xx adp94xx
    c:\windows\system32\drivers\adp94xx.sys
Kernel Driver No Disabled
Stopped OK Normal No No

adpahci adpahci
    c:\windows\system32\drivers\adpahci.sys
Kernel Driver No Disabled
Stopped OK Normal No No

adpu160m adpu160m
    c:\windows\system32\drivers\adpu160m.sys
Kernel Driver No Disabled
Stopped OK Normal No No

adpu320 adpu320
    c:\windows\system32\drivers\adpu320.sys
Kernel Driver No Disabled
Stopped OK Normal No No

afd Ancillary Function Driver for Winsock
    c:\windows\system32\drivers\afd.sys
Kernel Driver Yes System
Running OK Normal No Yes

agp440 Intel AGP Bus Filter
    c:\windows\system32\drivers\agp440.sys
Kernel Driver No Manual
Stopped OK Normal No No

```

```

aic78xx aic78xx
    c:\windows\system32\drivers\djsvs.sys
Kernel Driver No Disabled
Stopped OK Normal No No

aliide aliide
    c:\windows\system32\drivers\aliide.sys
Kernel Driver No Disabled
Stopped OK Critical No No

amdide amdide
    c:\windows\system32\drivers\amdide.sys
Kernel Driver No Disabled
Stopped OK Critical No No

amd64 AMD K8 Processor Driver
    c:\windows\system32\drivers\amd64.sys
Kernel Driver No Disabled
Stopped OK Normal No No

arc arc
    c:\windows\system32\drivers\arc.sys
Kernel Driver No Disabled
Stopped OK Normal No No

arcsas arcsas
    c:\windows\system32\drivers\arcsas.sys
Kernel Driver No Disabled
Stopped OK Normal No No

asynxmac RAS Asynchronous Media Driver
    c:\windows\system32\drivers\asynxmac.sys
Kernel Driver No Manual
Stopped OK Normal No No

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\b06bdrv.sys
Kernel Driver No Disabled
Stopped OK Normal No No

blbdrive blbdrive
    c:\windows\system32\drivers\blbdrive.sys
Kernel Driver No Disabled
Stopped OK Normal No No

browser Browser
    c:\windows\system32\drivers\browser.sys
File System Driver Yes Manual
Running OK Normal No Yes

brfiltlo Brother USB Mass-Storage Lower Filter
Driver
    c:\windows\system32\drivers\brfiltlo.sys
Kernel Driver No Manual
Stopped OK Normal No No

brfiltup Brother USB Mass-Storage Upper Filter
Driver
    c:\windows\system32\drivers\brfiltup.sys

```

	Kernel Driver	No	Manual		
	Stopped	OK	Normal	No	No
brserid (WDM)	Brother MFC Serial Port Interface Driver				
	c:\windows\system32\drivers\brserid.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
brserwdm	Brother WDM Serial driver				
	c:\windows\system32\drivers\brserwdm.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
brusbmdm	Brother MFC USB Fax Only Modem				
	c:\windows\system32\drivers\brusbmdm.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
brusbser	Brother MFC USB Serial WDM Driver				
	c:\windows\system32\drivers\brusbser.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Normal	No	No
cdfs	CD/DVD File System Reader				
	c:\windows\system32\drivers\cdfs.sys				
	File System Driver	Yes	Disabled		
	Running	OK	Normal	No	Yes
cdrom	CD-ROM Driver				
	c:\windows\system32\drivers\cdrom.sys				
	Kernel Driver	Yes	System		
	Running	OK	Normal	No	Yes
circlass	Consumer IR Devices				
	c:\windows\system32\drivers\circlass.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
clfs	Common Log (CLFS)				
	c:\windows\system32\clfs.sys	Kernel Driver			
	Yes	Boot	Running	OK	
	Critical	No	Yes		
cmdide	cmdide				
	c:\windows\system32\drivers\cmdide.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Critical	No	No
compbatt	Microsoft Composite Battery Driver				
	c:\windows\system32\drivers\compbatt.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Critical	No	No
crdisk	Crcdisk Filter Driver				
	c:\windows\system32\drivers\crdisk.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
csc	Offline Files Driver				
	c:\windows\system32\drivers\csc.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No

dfsc	DFS Namespace Client Driver				
	c:\windows\system32\drivers\dfsc.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
dfsdriver	DFS Namespace Server Filter Driver				
	c:\windows\system32\drivers\dfs.sys				
	File System Driver	Yes	System		
	Running	OK	Normal	No	Yes
disk	Disk Driver				
	c:\windows\system32\drivers\disk.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
dxgkrnl	LDDM Graphics Subsystem				
	c:\windows\system32\drivers\dxgkrnl.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Ignore	No	No
elexpress	Intel(R) PRO/1000 PCI Express Network Connection Driver				
	c:\windows\system32\drivers\ele6032e.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Normal	No	Yes
elg60	Intel(R) PRO/1000 NDIS 6 Adapter Driver				
	c:\windows\system32\drivers\elg6032e.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Normal	No	No
elxstor	elxstor				
	c:\windows\system32\drivers\elxstor.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
errdev	Microsoft Hardware Error Device Driver				
	c:\windows\system32\drivers\errdev.sys				
	Kernel Driver	No	Disabled		
	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\fastfat.sys				
	File System Driver	No	Manual		
	Stopped	OK	Normal	No	No
fdc	Floppy Disk Controller Driver				
	c:\windows\system32\drivers\fdc.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
fileinfo	File Information FS MiniFilter				
	c:\windows\system32\drivers\fileinfo.sys				
	File System Driver	No	Manual		
	Stopped	OK	Normal	No	No

filetrace	FileTrace				
	c:\windows\system32\drivers\filetrace.sys				
	File System Driver	No	Manual		
	Stopped	OK	Normal	No	No
flpydisk	Floppy Disk Driver				
	c:\windows\system32\drivers\flpydisk.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
fltmgr	FltMgr				
	c:\windows\system32\drivers\fltmgr.sys				
	File System Driver	Yes	Boot		
	Running	OK	Critical	No	Yes
gagp30kx	Microsoft Generic AGPv3.0 Filter for K8 Processors				
	c:\windows\system32\drivers\gagp30kx.sys				
	Kernel Driver	No	Manual		
	Stopped	OK	Normal	No	No
hdaudbus	Microsoft UAA Bus Driver for High Definition Audio				
	c:\windows\system32\drivers\hdaudbus.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Normal	No	No
hidbth	Microsoft Bluetooth HID Miniport				
	c:\windows\system32\drivers\hidbth.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Ignore	No	No
hidir	Microsoft Infrared HID Driver				
	c:\windows\system32\drivers\hidir.sys				
	Kernel Driver	No	Disabled		
	Stopped	OK	Ignore	No	No
hidusb	Microsoft HID Class Driver				
	c:\windows\system32\drivers\hidusb.sys				
	Kernel Driver	Yes	Manual		
	Running	OK	Ignore	No	Yes
hpcisss	HpCISSs				
	c:\windows\system32\drivers\hpcisss.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
hpcisss2	HpCISSs2				
	c:\windows\system32\drivers\hpcisss2.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
hpgcissb	Smart Array Controllers Non-Miniport Bus Driver				
	c:\windows\system32\drivers\hpgcissb.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes
hpgcissd	Smart Array Controllers Non-Miniport Disk Driver				
	c:\windows\system32\drivers\hpgcissd.sys				
	Kernel Driver	Yes	Boot		
	Running	OK	Normal	No	Yes

http	HTTP c:\windows\system32\drivers\http.sys Kernel Driver Yes Manual Running OK Normal No Yes	Kernel Driver No Disabled Stopped OK Critical No No	Running OK Normal No Yes
i2omp	i2omp c:\windows\system32\drivers\i2omp.sys Kernel Driver No Disabled Stopped OK Normal No No	iscsiprt iScsiPort Driver c:\windows\system32\drivers\msiscsi.sys Kernel Driver Yes Manual Running OK Normal No Yes	megasas megasas c:\windows\system32\drivers\megasas.sys Kernel Driver No Disabled Stopped OK Normal No No
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver c:\windows\system32\drivers\i8042prt.sys Kernel Driver Yes System Running OK Normal No Yes	iteatapi ITEATAPI_Service_Install c:\windows\system32\drivers\iteatapi.sys Kernel Driver No Disabled Stopped OK Normal No No	megasr MegaSR c:\windows\system32\drivers\megasr.sys Kernel Driver No Disabled Stopped OK Normal No No
iastorv	Intel RAID Controller Vista c:\windows\system32\drivers\iastorv.sys Kernel Driver No Disabled Stopped OK Normal No No	iteraid ITERAID_Service_Install c:\windows\system32\drivers\iteraid.sys Kernel Driver No Disabled Stopped OK Normal No No	modem Modem c:\windows\system32\drivers\modem.sys Kernel Driver No Manual Stopped OK Ignore No No
iirsp	iirsp c:\windows\system32\drivers\iirsp.sys Kernel Driver No Disabled Stopped OK Normal No No	kbdclass Keyboard Class Driver c:\windows\system32\drivers\kbdclass.sys Kernel Driver Yes System Running OK Normal No Yes	monitor Microsoft Monitor Class Function Driver Service c:\windows\system32\drivers\monitor.sys Kernel Driver Yes Manual Running OK Normal No Yes
intelide	intelide c:\windows\system32\drivers\intelide.sys Kernel Driver No Disabled Stopped OK Critical No No	kbdhid Keyboard HID Driver c:\windows\system32\drivers\kbdhid.sys Kernel Driver Yes System Running OK Ignore No Yes	mouclass Mouse Class Driver c:\windows\system32\drivers\mouclass.sys Kernel Driver Yes System Running OK Normal No Yes
intelppm	Intel Processor Driver c:\windows\system32\drivers\intelppm.sys Kernel Driver Yes Manual Running OK Normal No Yes	ksecdd KSecDD c:\windows\system32\drivers\ksecdd.sys Kernel Driver No Boot Running OK Critical No Yes	mouhid Mouse HID Driver c:\windows\system32\drivers\mouhid.sys Kernel Driver Yes Manual Running OK Ignore No Yes
ioatdma	Intel(R) QuickData Technology Device c:\windows\system32\drivers\qd260x64.sys Kernel Driver No Disabled Stopped OK Normal No No	ksthunk Kernel Streaming Thunks c:\windows\system32\drivers\ksthunk.sys Kernel Driver No Manual Stopped OK Normal No No	mountmgr Mount Point Manager c:\windows\system32\drivers\mountmgr.sys Kernel Driver Yes Boot Running OK Critical No Yes
ipfilterdriver	IP Traffic Filter Driver c:\windows\system32\drivers\ipfltdrv.sys Kernel Driver No Manual Stopped OK Normal No No	lldio Link-Layer Topology Discovery Mapper I/O Driver c:\windows\system32\drivers\lldio.sys Kernel Driver Yes Auto Running OK Normal No Yes	mpio Microsoft Multi-Path Bus Driver c:\windows\system32\drivers\mpio.sys Kernel Driver No Disabled Stopped OK Normal No No
ipmidrv	IPMIDRV c:\windows\system32\drivers\ipmidrv.sys Kernel Driver Yes Manual Running OK Normal No Yes	lsi_fc LSI_FC c:\windows\system32\drivers\lsi_fc.sys Kernel Driver No Disabled Stopped OK Normal No No	mpsdrv Windows Firewall Authorization Driver c:\windows\system32\drivers\mpsdrv.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	lsi_sas LSI_SAS c:\windows\system32\drivers\lsi_sas.sys Kernel Driver No Disabled Stopped OK Normal No No	mrraid35x Mrraid35x c:\windows\system32\drivers\mrraid35x.sys Kernel Driver No Disabled Stopped OK Normal No No
irenum	IR Bus Enumerator c:\windows\system32\drivers\irenum.sys Kernel Driver No Manual Stopped OK Ignore No No	lsi_scsi LSI_SCSI c:\windows\system32\drivers\lsi_scsi.sys Kernel Driver No Disabled Stopped OK Normal No No	mrx smb SMB MiniRedirector Wrapper and Engine c:\windows\system32\drivers\mrx smb.sys File System Driver Yes Manual Running OK Normal No Yes
isapnp	PnP ISA/EISA Bus Driver c:\windows\system32\drivers\isapnp.sys	luafv UAC File Virtualization c:\windows\system32\drivers\luafv.sys File System Driver Yes Auto	mrx smb10 SMB 1.x MiniRedirector c:\windows\system32\drivers\mrx smb10.sys File System Driver Yes Manual Running OK Normal No Yes

mrxsmb20	SMB 2.0 MiniRedirector c:\windows\system32\drivers\mrxsmb20.sys File System Driver Yes Manual Running OK Normal No Yes	Kernel Driver Yes Manual Running OK Normal No Yes	Stopped OK Normal No No
msahci	msahci c:\windows\system32\drivers\msahci.sys Kernel Driver No Disabled Stopped OK Critical No No	NetBIOS Interface c:\windows\system32\drivers\netbios.sys File System Driver Yes System Running OK Normal No Yes	ohcil394 NEC FireWarden OHCI Compliant IEEE 1394 Host Controller c:\windows\system32\drivers\ohcil394.sys Kernel Driver No Disabled Stopped OK Normal No No
msdsm	Microsoft Multi-Path Device Specific Module c:\windows\system32\drivers\msdsm.sys Kernel Driver No Disabled Stopped OK Normal No No	NETBT c:\windows\system32\drivers\netbt.sys Kernel Driver Yes System Running OK Normal No Yes	parport Parallel port driver c:\windows\system32\drivers\parport.sys Kernel Driver No Disabled Stopped OK Normal No No
msfs	Msfs c:\windows\system32\drivers\msfs.sys File System Driver Yes System Running OK Normal No Yes	nfrd960 c:\windows\system32\drivers\nfrd960.sys Kernel Driver No Disabled Stopped OK Normal No No	partmgr Partition Manager c:\windows\system32\drivers\partmgr.sys Kernel Driver Yes Boot Running OK Critical No Yes
msisadrv	ISA/EISA Class Driver c:\windows\system32\drivers\msisadrv.sys Kernel Driver Yes Boot Running OK Critical No Yes	npfs c:\windows\system32\drivers\npfs.sys File System Driver Yes System Running OK Normal No Yes	pci PCI Bus Driver c:\windows\system32\drivers\pci.sys Kernel Driver Yes Boot Running OK Critical No Yes
msrpc	MsRPC c:\windows\system32\drivers\msrpc.sys Kernel Driver No Manual Stopped OK Normal No No	nsiproxy NSI proxy service c:\windows\system32\drivers\nsiproxy.sys Kernel Driver Yes System Running OK Normal No Yes	pciide pciide c:\windows\system32\drivers\pciide.sys Kernel Driver Yes Boot Running OK Critical No Yes
msmbios	Microsoft System Management BIOS Driver c:\windows\system32\drivers\msmbios.sys Kernel Driver Yes Manual Running OK Normal No Yes	ntfs c:\windows\system32\drivers\ntfs.sys File System Driver Yes Manual Running OK Normal No Yes	pcmcia pcmcia c:\windows\system32\drivers\pcmcia.sys Kernel Driver No Disabled Stopped OK Normal No No
mup	Mup c:\windows\system32\drivers\mup.sys File System Driver Yes Boot Running OK Normal No Yes	null c:\windows\system32\drivers\null.sys Kernel Driver Yes System Running OK Normal No Yes	peauth PEAUTH c:\windows\system32\drivers\peauth.sys Kernel Driver Yes Auto Running OK Normal No Yes
ndis	NDIS System Driver c:\windows\system32\drivers\ndis.sys Kernel Driver Yes Boot Running OK Critical No Yes	nvraid NVIDIA nForce RAID Driver c:\windows\system32\drivers\nvraid.sys Kernel Driver No Disabled Stopped OK Normal No No	plxsvc PLX PCI/PCIe Service Driver c:\windows\system32\drivers\plxsvc.sys Kernel Driver Yes Auto Running OK Ignore No Yes
ndistapi	Remote Access NDIS TAPI Driver c:\windows\system32\drivers\ndistapi.sys Kernel Driver Yes Manual Running OK Normal No Yes	nvstor c:\windows\system32\drivers\nvstor.sys Kernel Driver No Disabled Stopped OK Critical No No	pptpminiport WAN Miniport (PPTP) c:\windows\system32\drivers\rasppptp.sys Kernel Driver Yes Manual Running OK Normal No Yes
ndisuio	NDIS Usermode I/O Protocol c:\windows\system32\drivers\ndisuio.sys Kernel Driver No Manual Stopped OK Normal No No	nv_agp NVIDIA nForce AGP Bus Filter c:\windows\system32\drivers\nv_agp.sys Kernel Driver No Manual Stopped OK Normal No No	processor Processor Driver c:\windows\system32\drivers\processr.sys Kernel Driver No Disabled Stopped OK Normal No No
ndiswan	Remote Access NDIS WAN Driver c:\windows\system32\drivers\ndiswan.sys Kernel Driver Yes Manual Running OK Normal No Yes	nxnd5hp HP Multifunction 1/10 Gigabit Server Adapter c:\windows\system32\drivers\hpn5x64.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
ndproxy	NDIS Proxy c:\windows\system32\drivers\ndproxy.sys	nxnd6hp HP Multifunctions 1/10 Gigabit Server Adapter c:\windows\system32\drivers\hpn6x64.sys Kernel Driver No Manual	ql2300 QLogic Fibre Channel STOR Miniport Driver (wx64 IP) c:\windows\system32\drivers\ql2300.sys Kernel Driver Yes Boot

	Running	OK	Normal	No	Yes
ql40xx	QLogic iSCSI Miniport Driver c:\windows\system32\drivers\ql40xx.sys Kernel Driver No Disabled Stopped OK Normal No No				
rasacd	Remote Access Auto Connection Driver c:\windows\system32\drivers\rasacd.sys Kernel Driver Yes System Running OK Normal No Yes				
rasl2tp	WAN Miniport (L2TP) c:\windows\system32\drivers\rasl2tp.sys Kernel Driver Yes Manual Running OK Normal No Yes				
rasppoe	Remote Access PPOE Driver c:\windows\system32\drivers\rasppoe.sys Kernel Driver Yes Manual Running OK Normal No Yes				
rassttp	WAN Miniport (SSTP) c:\windows\system32\drivers\rassttp.sys Kernel Driver Yes Manual Running OK Normal No Yes				
rdbss	Redirected Buffering Sub Sysytem c:\windows\system32\drivers\rdbss.sys File System Driver Yes System Running OK Normal No Yes				
rdpcdd	RDPCDD c:\windows\system32\drivers\rdpcdd.sys Kernel Driver Yes System Running OK Ignore No Yes				
rdpdr	Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys Kernel Driver Yes Manual Running OK Normal No Yes				
rdpenccd	RDP Encoder Mirror Driver c:\windows\system32\drivers\rdpenccd.sys Kernel Driver Yes System Running OK Ignore No Yes				
rdpwd	RDP Winstation Driver c:\windows\system32\drivers\rdpwd.sys Kernel Driver Yes Manual Running OK Ignore No Yes				
rspnдр	Link-Layer Topology Discovery Responder c:\windows\system32\drivers\rspnдр.sys Kernel Driver Yes Auto Running OK Normal No Yes				
s3cap	Microsoft Emulated S3 Device Cap Driver c:\windows\system32\drivers\s3cap.sys Kernel Driver No Disabled Stopped OK Normal No No				

sacdrv	sacdrv c:\windows\system32\drivers\sacdrv.sys Kernel Driver No Boot Stopped OK Ignore No No				
sbp2port	SBP-2 Transport/Protocol Bus Driver c:\windows\system32\drivers\sbp2port.sys Kernel Driver No Disabled Stopped OK Normal No No				
secdrv	Security Driver c:\windows\system32\drivers\secdrv.sys Kernel Driver Yes Auto Running OK Normal No Yes				
serenum	Serenum Filter Driver c:\windows\system32\drivers\serenum.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\sermouse.sys Kernel Driver No Disabled Stopped OK Normal No No				
sffdisk	SFF Storage Class Driver c:\windows\system32\drivers\sffdisk.sys Kernel Driver No Disabled Stopped OK Normal No No				
sffp_mmc	SFF Storage Protocol Driver for MMC c:\windows\system32\drivers\sffp_mmc.sys Kernel Driver No Manual Stopped OK Normal No No				
sffp_sd	SFF Storage Protocol Driver for SDBus c:\windows\system32\drivers\sffp_sd.sys Kernel Driver No Manual Stopped OK Normal No No				
sfloppy	High-Capacity Floppy Disk Drive c:\windows\system32\drivers\sfloppy.sys Kernel Driver No Manual Stopped OK Normal No No				
sisraid2	SiSRaid2 c:\windows\system32\drivers\sisraid2.sys Kernel Driver No Disabled Stopped OK Normal No No				
sisraid4	SiSRaid4 c:\windows\system32\drivers\sisraid4.sys Kernel Driver No Disabled Stopped OK Normal No No				
smb	Message-oriented TCP/IP and TCP/IPv6 Protocol (SMB session)				

	c:\windows\system32\drivers\smb.sys Kernel Driver Yes System Running OK Normal No Yes				
spldr	Security Processor Loader Driver c:\windows\system32\drivers\spldr.sys Kernel Driver Yes Boot Running OK Critical No Yes				
srv	srv c:\windows\system32\drivers\srv.sys File System Driver Yes Manual Running OK Normal No Yes				
srv2	srv2 c:\windows\system32\drivers\srv2.sys File System Driver Yes Manual Running OK Normal No Yes				
srvnet	srvnet c:\windows\system32\drivers\srvnet.sys File System Driver Yes Manual Running OK Normal No Yes				
storflt	Disk VMBUS Acceleration Filter Driver c:\windows\system32\drivers\storflt.sys Kernel Driver Yes Boot Running OK Normal No Yes				
storvsc	storvsc c:\windows\system32\drivers\storvsc.sys Kernel Driver No Disabled Stopped OK Normal No No				
storvsp	Microsoft Virtual Disk Server Driver c:\windows\system32\drivers\storvsp.sys Kernel Driver No Disabled Stopped OK Normal No No				
swenum	Software Bus Driver c:\windows\system32\drivers\swenum.sys Kernel Driver Yes Manual Running OK Normal No Yes				
symc8xx	Symc8xx c:\windows\system32\drivers\symc8xx.sys Kernel Driver No Disabled Stopped OK Normal No No				
sym_hi	Sym_hi c:\windows\system32\drivers\sym_hi.sys Kernel Driver No Disabled Stopped OK Normal No No				
sym_u3	Sym_u3 c:\windows\system32\drivers\sym_u3.sys Kernel Driver No Disabled Stopped OK Normal No No				
tcpip	TCP/IP Protocol Driver c:\windows\system32\drivers\tcpip.sys Kernel Driver Yes Boot				

	Running	OK	Normal	No	Yes		ulsata	Ulsata c:\windows\system32\drivers\ulsata.sys Kernel Driver No Disabled Stopped OK Normal No No		vga	vga c:\windows\system32\drivers\vgapnp.sys Kernel Driver Yes Manual Running OK Ignore No Yes
tcpip6	Microsoft IPv6 Protocol Driver c:\windows\system32\drivers\tcpip.sys Kernel Driver No Manual Stopped OK Normal No No						ulsata2	ulsata2 c:\windows\system32\drivers\ulsata2.sys Kernel Driver No Disabled Stopped OK Normal No No		vgasave	VgaSave c:\windows\system32\drivers\vga.sys Kernel Driver Yes System Running OK Ignore No Yes
tcpipreg	TCP/IP Registry Compatibility c:\windows\system32\drivers\tcpipreg.sys Kernel Driver Yes Auto Running OK Normal No Yes						umbus	UMBus Enumerator Driver c:\windows\system32\drivers\umbus.sys Kernel Driver Yes Manual Running OK Normal No Yes		viaide	viaide c:\windows\system32\drivers\viaide.sys Kernel Driver No Disabled Stopped OK Critical No No
tdpipe	TDPIPE c:\windows\system32\drivers\tdpipe.sys Kernel Driver No Manual Stopped OK Normal No No						umpass	Microsoft UMPass Driver c:\windows\system32\drivers\umpass.sys Kernel Driver No Disabled Stopped OK Normal No No		vid	Virtualization Infrastructure Driver c:\windows\system32\drivers\vid.sys Kernel Driver No Disabled Stopped OK Normal No No
tdtcp	TDTCP c:\windows\system32\drivers\tdtcp.sys Kernel Driver Yes Manual Running OK Normal No Yes						usbccgp	Microsoft USB Generic Parent Driver c:\windows\system32\drivers\usbccgp.sys Kernel Driver Yes Manual Running OK Normal No Yes		vmbus	VMBus c:\windows\system32\drivers\vmbus.sys Kernel Driver No Disabled Stopped OK Normal No No
tdx	NetIO Legacy TDI Support Driver c:\windows\system32\drivers\tdx.sys Kernel Driver Yes System Running OK Normal No Yes						usbcir	eHome Infrared Receiver (USB CIR) c:\windows\system32\drivers\usbcir.sys Kernel Driver No Disabled Stopped OK Normal No No		volmgr	Volume Manager Driver c:\windows\system32\drivers\volmgr.sys Kernel Driver Yes Boot Running OK Critical No Yes
termdd	Terminal Device Driver c:\windows\system32\drivers\termdd.sys Kernel Driver Yes System Running OK Normal No Yes						usbhci	Microsoft USB 2.0 Enhanced Host Controller Miniport Driver c:\windows\system32\drivers\usbhci.sys Kernel Driver Yes Manual Running OK Normal No Yes		volmgrx	Dynamic Volume Manager c:\windows\system32\drivers\volmgrx.sys Kernel Driver Yes Boot Running OK Critical No Yes
tssecsrv	Terminal Services Security Filter Driver c:\windows\system32\drivers\tssecsrv.sys Kernel Driver Yes Manual Running OK Ignore No Yes						usbhub	USB2 Enabled Hub c:\windows\system32\drivers\usbhub.sys Kernel Driver Yes Manual Running OK Normal No Yes		volsnap	Storage volumes c:\windows\system32\drivers\volsnap.sys Kernel Driver Yes Boot Running OK Critical No Yes
tunnel Driver	Microsoft IPv6 Tunnel Miniport Adapter c:\windows\system32\drivers\tunnel.sys Kernel Driver Yes Manual Running OK Normal No Yes						usbohci	Microsoft USB Open Host Controller Miniport Driver c:\windows\system32\drivers\usbohci.sys Kernel Driver No Disabled Stopped OK Normal No No		vsmraid	vsmraid c:\windows\system32\drivers\vsmraid.sys Kernel Driver No Disabled Stopped OK Normal No No
uagp35	Microsoft AGPv3.5 Filter c:\windows\system32\drivers\uagp35.sys Kernel Driver No Manual Stopped OK Normal No No						usbprint	Microsoft USB PRINTER Class c:\windows\system32\drivers\usbprint.sys Kernel Driver No Disabled Stopped OK Normal No No		wacompen	Wacom Serial Pen HID Driver c:\windows\system32\drivers\wacompen.sys Kernel Driver No Disabled Stopped OK Normal No No
udfs	udfs c:\windows\system32\drivers\udfs.sys File System Driver No Disabled Stopped OK Normal No No						usbstor	USB Mass Storage Driver c:\windows\system32\drivers\usbstor.sys Kernel Driver No Manual Stopped OK Normal No No		wanarp	Remote Access IP ARP Driver c:\windows\system32\drivers\wanarp.sys Kernel Driver No Manual Stopped OK Normal No No
uliagpkx	Uli AGP Bus Filter c:\windows\system32\drivers\uliagpkx.sys Kernel Driver No Manual Stopped OK Normal No No						usbuhci	Microsoft USB Universal Host Controller Miniport Driver c:\windows\system32\drivers\usbuhci.sys Kernel Driver Yes Manual Running OK Normal No Yes		wanarpv6	Remote Access IPv6 ARP Driver c:\windows\system32\drivers\wanarpv6.sys Kernel Driver Yes System Running OK Normal No Yes
uliahci	uliahci c:\windows\system32\drivers\uliahci.sys Kernel Driver No Disabled Stopped OK Normal No No									wd	Microsoft Watchdog Timer Driver c:\windows\system32\drivers\wd.sys

	Kernel Driver	No	Disabled	
	Stopped	OK	Normal	No No
wdf01000	Kernel Mode Driver Frameworks service			
	c:\windows\system32\drivers\wdf01000.sys			
	Kernel Driver	Yes	Boot	
	Running	OK	Normal	No Yes
wmiacpi	Microsoft Windows Management Interface for			
ACPI	c:\windows\system32\drivers\wmiacpi.sys			
	Kernel Driver	No	Disabled	
	Stopped	OK	Normal	No No
ws2ifsl	Winsock IFS driver			
	c:\windows\system32\drivers\ws2ifsl.sys			
	Kernel Driver	No	Disabled	
	Stopped	OK	Normal	No No
[Signed Drivers]				
Device Name	Signed	Device Class		
	Driver Version	Driver Date		
	Manufacturer	INF Name	Driver Name	
	Device ID			
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&LDM#{711FD997-C4AC-11DE-BE03-0018FE2E1EBE}			
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&LDM#{711FD995-C4AC-11DE-BE03-0018FE2E1EBE}			
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE220A22			
0A0FFSET4000LENGTH8787EC000				
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21			
3DOFFSET7E00LENGTH17357D4E000				
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21			
3COFFSET20000LENGTH752C00000				
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21			
3FOFFSET20000LENGTH1D8A500000				
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		
	volume.inf	Not Available		
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21			
3EOFFSET20000LENGTH1A1F300000				
Generic volume	Yes	VOLUME		
	6.0.6002.16670	6/21/2006 Microsoft		

	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
010FFSET20000LENGTH2980800000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
1BOFFSET7E00LENGTH17357D4E000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
1A0FFSET20000LENGTH752C00000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
1DOFFSET20000LENGTH1D8A500000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
1COFFSET20000LENGTH1A1F300000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE10FFSE		
T20000LENGTH2980800000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
EFOFFSET7E00LENGTH17357D4E000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
EEOFFSET20000LENGTH752C00000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
FIOFFSET20000LENGTH1D8A500000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
FO0FFSET20000LENGTH1A1F300000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E5OFFSET7E00LENGTH17357D4E000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	

	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E40FFSET20000LENGTH752C00000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E70FFSET20000LENGTH1D8A500000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E60FFSET20000LENGTH1A1F300000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E90FFSET20000LENGTH2980800000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
E80FFSET7E00LENGTH17357D4E000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
EBOFFSET20000LENGTH752C00000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
EAOFFSET20000LENGTH1D8A500000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
EDOFFSET20000LENGTH1A1F300000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
ECOFFSET20000LENGTH2980800000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
000FFSET7E00LENGTH17357D4E000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
030FFSET20000LENGTH752C00000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
020FFSET20000LENGTH1D8A500000			
Generic volume	Yes	VOLUME	
	6.0.6002.16670	6/21/2006 Microsoft	
	volume.inf	Not Available	
	STORAGE\OLUME\1&19F7E59C&0&SIGNATURE6EFB21		
050FFSET20000LENGTH1A1F300000			

UMBus Enumerator	Yes	SYSTEM	Available	Not Available	Not Available	Not Available	Available	Not Available	ROOT\LEGACY_RDPCCD\0000
6.0.6001.18000	6/21/2006	Microsoft	Available	ROOT\LEGACY_WANARPV6\0000					
umbus.inf	Not Available		Dynamic Volume Manager	LEGACYDRIVER	Not Available	Not Available	Remote Access Auto Connection Driver	LEGACYDRIVER	Not Available
UMB\UMB\1&841921D&0&TSBUS			LEGACYDRIVER	Not Available	Not Available	Not Available	LEGACYDRIVER	Not Available	Not Available
UMBus Root Bus Enumerator	Yes	SYSTEM	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	ROOT\LEGACY_VOLMGRX\0000			Available	ROOT\LEGACY_RASACD\0000	
umbus.inf	Not Available		VgaSave	LEGACYDRIVER	Not Available	Not Available	QoS Packet Scheduler	LEGACYDRIVER	Not Available
ROOT\UMBUS\0000			Available	Not Available	Not Available	Not Available	LEGACYDRIVER	Not Available	Not Available
Microsoft System Management BIOS Driver	Yes		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
SYSTEM	6.0.6002.16670	6/21/2006	Available	ROOT\LEGACY_VGASAVE\0000			Available	ROOT\LEGACY_PSCHE\0000	
(Standard system devices)	machine.inf		Terminal Services Security Filter Driver	LEGACYDRIVER	Not Available	Not Available	PLX PCI/PCIe Service Driver	LEGACYDRIVER	Not Available
Not Available	ROOT\SYSTEM\0002		Available	LEGACYDRIVER	Not Available	Not Available	LEGACYDRIVER	Not Available	Not Available
Plug and Play Software Device Enumerator	Yes		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
SYSTEM	6.0.6002.16670	6/21/2006	Available	ROOT\LEGACY_TSSECSRV\0000			Available	ROOT\LEGACY_PLXSVC\0000	
(Standard system devices)	machine.inf		NetIO Legacy TDI Support Driver	LEGACYDRIVER	Not Available	Not Available	PEAUTH	LEGACYDRIVER	Not Available
Not Available	ROOT\SYSTEM\0000		Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
Terminal Server Mouse Driver	Yes	SYSTEM	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6002.16670	6/21/2006	(Standard system devices)	Available	ROOT\LEGACY_TDX\0000			Available	Not Available	ROOT\LEGACY_PEAUTH\0000
machine.inf	Not Available		TDTCP	LEGACYDRIVER	Not Available	Not Available	Null	Not Available	LEGACYDRIVER
ROOT\RDP_MOU\0000			Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
Terminal Server Keyboard Driver	Yes		Available	Not Available	Not Available	Not Available	Available	Not Available	ROOT\LEGACY_NULL\0000
SYSTEM	6.0.6002.16670	6/21/2006	TCP/IP Registry Compatibility	LEGACYDRIVER	Not Available	Not Available	NSI proxy service	LEGACYDRIVER	Not Available
(Standard system devices)	machine.inf		Available	LEGACYDRIVER	Not Available	Not Available	Not Available	LEGACYDRIVER	Not Available
Not Available	ROOT\RDP_KBD\0000		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
Terminal Server Device Redirector	Yes	SYSTEM	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6002.16670	6/21/2006	(Standard system devices)	Available	ROOT\LEGACY_TCIPREG\0000			Available	ROOT\LEGACY_NSIPROXY\0000	
machine.inf	Not Available		TCP/IP Protocol Driver	LEGACYDRIVER	Not Available	Not Available	NETBT	LEGACYDRIVER	Not Available
ROOT\RDPDR\0000			Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
WAN Miniport (SSTP)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	ROOT\LEGACY_NETBT\0000
6.0.6001.18000	6/21/2006	Microsoft	Available	ROOT\LEGACY_TCPIP\0000			Available	Not Available	Not Available
netssstp.inf	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
ROOT\MS_SSTP\MINIPORT\0000			Disk VMBUS Acceleration Filter Driver	LEGACYDRIVER	Not Available	Not Available	NDProxy	LEGACYDRIVER	Not Available
WAN Miniport (PPTP)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	ROOT\LEGACY_STORFLT\0000			Available	Not Available	Not Available
netrasa.inf	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
ROOT\MS_PPTP\MINIPORT\0000			Security Processor Loader Driver	LEGACYDRIVER	Not Available	Not Available	NDIS System Driver	LEGACYDRIVER	Not Available
WAN Miniport (PPPOE)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	ROOT\LEGACY_SPLDR\0000			Available	Not Available	Not Available
netrasa.inf	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
ROOT\MS_PPPOE\MINIPORT\0000			Message-oriented TCP/IP and TCP/IPV6 Protocol (SMB session)	LEGACYDRIVER	Not Available	Not Available	ISA/EISA Class Driver	LEGACYDRIVER	Not Available
WAN Miniport (IPv6)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
netrasa.inf	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
ROOT\MS_NDISWANIPV6\0000			Security Driver	LEGACYDRIVER	Not Available	Not Available	Available	ROOT\LEGACY_MSISADRV\0000	
WAN Miniport (IP)	Yes	NET	Available	Not Available	Not Available	Not Available	Windows Firewall Authorization Driver	LEGACYDRIVER	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
netrasa.inf	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
ROOT\MS_NDISWANIP\0000			Link-Layer Topology Discovery Responder	LEGACYDRIVER	Not Available	Not Available	Mount Point Manager	LEGACYDRIVER	Not Available
WAN Miniport (Network Monitor)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
netrasa.inf	Not Available		Available	ROOT\LEGACY_RSPNDR\0000			Available	Not Available	Not Available
ROOT\MS_NDISWANBH\0000			RDP Winstation Driver	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
WAN Miniport (L2TP)	Yes	NET	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
6.0.6001.18000	6/21/2006	Microsoft	Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
netrasa.inf	Not Available		Available	ROOT\LEGACY_RDPWD\0000			Available	Not Available	Not Available
ROOT\MS_L2TP\MINIPORT\0000			RDP Encoder Mirror Driver	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
Kernel Mode Driver Frameworks service	Not Available		Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
LEGACYDRIVER	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
Available	Not Available		Available	ROOT\LEGACY_RDPENCDD\0000			HTTP	LEGACYDRIVER	Not Available
Available	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
Available	ROOT\LEGACY_WDF01000\0000		RDPCCD	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available
Remote Access IPv6 ARP Driver	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available
LEGACYDRIVER	Not Available		Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available

```

HpCISSs Not Available LEGACYDRIVER Not
Available Not Available Not Available Not
Available Not Available
ROOT\LEGACY_HPCISSS\0000
Crcdisk Filter Driver Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_CRCDISK\0000
Common Log (CLFS) Not Available LEGACYDRIVER
Not Available Not Available Not
Available Not Available Not Available
ROOT\LEGACY_CLFS\0000
Ancillary Function Driver for Winsock Not Available
LEGACYDRIVER Not Available Not
Available Not Available Not Available Not
Available ROOT\LEGACY_AFD\0000
Microsoft iSCSI Initiator Yes SCSIADAPTER
6.0.6002.16670 6/21/2006 Microsoft
iscsi.inf Not Available
ROOT\ISCSIPRT\0000
ACPI Fixed Feature Button Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\FIXEDBUTTON\2&DABA3FF&1
ACPI Thermal Zone Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\THERMALZONE\THM0
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_8086&DEV_3423&SUBSYS_000B003C&REV_1
3\3&13C0B0C5&1&A2
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_8086&DEV_3422&SUBSYS_000B003C&REV_1
3\3&13C0B0C5&1&A1
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_8086&DEV_342E&SUBSYS_000B003C&REV_1
3\3&13C0B0C5&1&A0
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3439&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&74
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_341E&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&72
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available

```

```

PCI\VEN_8086&DEV_341D&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&71
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_341C&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&70
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_341A&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&6E
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3419&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&6D
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3418&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&6C
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3417&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&6B
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_343C&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&6A
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_343B&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&69
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_343A&SUBSYS_00000000&REV_1
3\3&13C0B0C5&1&68
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3411&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&50
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&2
ADD7011&0&0400004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&2
ADD7011&0&0300004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available

```

```

HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&2
ADD7011&0&0200004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&2
ADD7011&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&2
ADD7011&0&0000004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&11FF4A0C&0&0048
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3410&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&48
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&3
40D2586&0&0400004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&3
40D2586&0&0300004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&3
40D2586&0&0200004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&3
40D2586&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&3
40D2586&0&0000004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&190F69AE&0&0040
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340F&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&40
Disk drive Yes DISKDRIVE
6.0.6002.16670 6/21/2006 (Standard
disk drives) disk.inf Not Available

```

```

SCSI\DISK&VEN_HP&PROD_MSA2324FC\5&2BAE6B5C&
0&000001
Generic SCSI Enclosure Device Yes SYSTEM
6.0.6001.18000 6/21/2006 Microsoft
scsidev.inf Not Available
SCSI\ENCLOSURE&VEN_HP&PROD_MSA2324FC\5&2BAE
6B5C&0&000000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.8.17 5/21/2009 QLogic oem11.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2B0041D6&0&0138
Disk drive Yes DISKDRIVE
6.0.6002.16670 6/21/2006 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_MSA2324FC\5&8ACC8FF&0
&000002
Generic SCSI Enclosure Device Yes SYSTEM
6.0.6001.18000 6/21/2006 Microsoft
scsidev.inf Not Available
SCSI\ENCLOSURE&VEN_HP&PROD_MSA2324FC\5&8ACC
8FF&0&000000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.8.17 5/21/2009 QLogic oem11.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_7041103C&REV_0
2\4&2B0041D6&0&0038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340E&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&38
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340D&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&30
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340C&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&28
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340B&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&20
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&8
CA42AE&0&0400004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&8
CA42AE&0&0300004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&3892F0CE&0&0008

```

```

Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&8
CA42AE&0&0200004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&8
CA42AE&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&8
CA42AE&0&0000004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&59E450C&0&0018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340A&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&18
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3409&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&10
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
0FB3843&0&0400004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
0FB3843&0&0300004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
0FB3843&0&0200004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
0FB3843&0&0000004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&3892F0CE&0&0008

```

```

PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3408&SUBSYS_330B103C&REV_1
3\3&13C0B0C5&1&08
PCI bus Yes SYSTEM 6.0.6002.16670
6/21/2006 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\1
CD-ROM Drive Yes CDROM
6.0.6002.16670 6/21/2006 (Standard CD-
ROM drives) cdrom.inf Not Available
IDE\CDROMHLL-DT-ST_DVD-
ROM_GDRH20N_____D8E4_____5&2C0B779D&0&1.0.
0
IDE Channel Yes HDC
6.0.6002.16670 6/21/2006 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&CF23815&0&1
IDE Channel Yes HDC
6.0.6002.16670 6/21/2006 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available
PCIIDE\IDECHANNEL\4&CF23815&0&0
Standard Dual Channel PCI IDE Controller Yes
HDC 6.0.6002.16670 6/21/2006
(Standard IDE ATA/ATAPI controllers)
mshdc.inf Not Available
PCI\VEN_8086&DEV_3A20&SUBSYS_330D103C&REV_0
0\3&267A616A&0&FA
System CMOS/real time clock Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\PNP0B00\4&1CFF2C4C&0
Communications Port Yes PORTS
6.0.6001.18000 6/21/2006 (Standard
port types) mports.inf Not Available
ACPI\PNP0501\0
Extended IO Bus Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\PNP0A06\4&1CFF2C4C&0
PS/2 Compatible Mouse Yes MOUSE
6.0.6001.18000 6/21/2006 Microsoft
msmouse.inf Not Available
ACPI\PNP0F13\4&1CFF2C4C&0
Standard PS/2 Keyboard Yes KEYBOARD
6.0.6002.16670 6/21/2006 (Standard
keyboards) keyboard.inf Not Available
ACPI\PNP0303\4&1CFF2C4C&0
System speaker Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\PNP0800\4&1CFF2C4C&0
Direct memory access controller Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
ACPI\PNP0200\4&1CFF2C4C&0
High precision event timer Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard

```

```

system devices) machine.inf Not Available
ACPI\PNP0103\0
System timer Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\PNP0100\4&1CFF2C4C&0
Microsoft Generic IPMI Compliant Device Yes
SYSTEM 6.0.6002.16670 6/21/2006
Microsoft ipmidrv.inf Not Available
ACPI\IPI0001\0
Motherboard resources Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
ACPI\PNP0C02\0
PCI standard ISA bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3A18&SUBSYS_00000000&REV_0
0\3&267A616A&0&F8
IPMI Interface Not Available Not Available
Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_0
0\4&10AF73B4&0&26F0
Generic USB Hub Yes USB
6.0.6002.16670 6/21/2006 (Generic USB
Hub) usb.inf Not Available
USB\VID_03F0&PID_1327\6&2657C2B7&0&2
HID-compliant mouse Yes MOUSE
6.0.6001.18000 6/21/2006 Microsoft
msmouse.inf Not Available
HID\VID_03F0&PID_1027&MI_01\8&1D06CA04&0&00
00
USB Human Interface Device Yes HIDCLASS
6.1.6002.16670 6/21/2006 (Standard
system devices) input.inf Not Available
USB\VID_03F0&PID_1027&MI_01\7&EEB5304&0&000
1
HID Keyboard Device Yes KEYBOARD
6.0.6002.16670 6/21/2006 (Standard
keyboards) keyboard.inf Not Available
HID\VID_03F0&PID_1027&MI_00\8&9D6ADC6&0&000
0
USB Human Interface Device Yes HIDCLASS
6.1.6002.16670 6/21/2006 (Standard
system devices) input.inf Not Available
USB\VID_03F0&PID_1027&MI_00\7&EEB5304&0&000
0
USB Composite Device Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usb.inf Not Available
USB\VID_03F0&PID_1027\6&2657C2B7&0&1
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB\5&2E18C0B0&0
Standard Universal PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_0
0\4&10AF73B4&0&24F0

```

```

Base System Device Not Available Not Available
Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_0
3\4&10AF73B4&0&22F0
Base System Device Not Available Not Available
Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_0
3\4&10AF73B4&0&20F0
Generic PnP Monitor Yes MONITOR
6.0.6001.18000 6/21/2006 (Standard
monitor types) monitor.inf Not Available
DISPLAY\AVO0000\5&34DBC65D&0&12345678&01&03
Standard VGA Graphics Adapter Yes DISPLAY
6.0.6001.18000 6/21/2006 (Standard
display types) display.inf Not Available
PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_0
2\4&10AF73B4&0&18F0
Intel(R) 82801 PCI Bridge - 244E Yes
SYSTEM 6.0.6002.16670 6/21/2006
Intel machine.inf Not Available
PCI\VEN_8086&DEV_244E&SUBSYS_330D103C&REV_9
0\3&267A616A&0&F0
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB20\4&33D8AB38&0
Standard Enhanced PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_3A3A&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EF
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&2829171E&0
Standard Universal PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_3A39&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EB
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&51FE5CF&0
Standard Universal PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_3A36&SUBSYS_330D103C&REV_0
0\3&267A616A&0&EA
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&EB0EDCD&0
Standard Universal PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)

```

```

usbport.inf Not Available
PCI\VEN_8086&DEV_3A35&SUBSYS_330D103C&REV_0
0\3&267A616A&0&E9
USB Root Hub Yes USB
6.0.6002.16670 6/21/2006 (Standard USB
Host Controller) usbport.inf Not Available
USB\ROOT_HUB\4&290F6BAB&0
Standard Universal PCI to USB Host Controller Yes
USB 6.0.6002.16670 6/21/2006
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_3A34&SUBSYS_330D103C&REV_0
0\3&267A616A&0&E8
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
9169F4&0&0400004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
9169F4&0&0300004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
9169F4&0&0200004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
9169F4&0&0100004000000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
9169F4&0&0000004000000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&294C1E27&0&00E0
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3A40&SUBSYS_330D103C&REV_0
0\3&267A616A&0&E0
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_8086&DEV_3423&SUBSYS_000B003C&REV_1
3\3&267A616A&0&A2
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available
PCI\VEN_8086&DEV_3422&SUBSYS_000B003C&REV_1
3\3&267A616A&0&A1
System Interrupt Controller Not Available Not
Available Not Available Not Available Not
Available Not Available Not Available

```



```

PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\5&1A9C2C35&0&600038
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4
81C591&0&0400000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4
81C591&0&0300000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4
81C591&0&0200000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4
81C591&0&0100000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&4
81C591&0&0000000400000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\6&179CC572&0&00500038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\5&1A9C2C35&0&500038
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&D
AEB7ED&0&0400000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&D
AEB7ED&0&0300000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&D
AEB7ED&0&0200000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&D
AEB7ED&0&0100000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available

```

```

HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&D
AEB7ED&0&0000000400000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\6&3B2C24D1&0&00480038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\5&1A9C2C35&0&480038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\5&1A9C2C35&0&400038
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1
657C522&0&0400000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1
657C522&0&0300000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1
657C522&0&0200000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1
657C522&0&0100000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\7&1
657C522&0&0000000400000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\6&16266DA6&0&00080038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\5&1A9C2C35&0&080038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_10B5&DEV_8548&SUBSYS_854810B5&REV_A
A\4&2B2265CA&0&0038

```

```

PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340E&SUBSYS_330B103C&REV_1
3\3&267A616A&0&38
Disk drive Yes DISKDRIVE
6.0.6002.16670 6/21/2006 (Standard
disk drives) disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1BEF
C3D4&0&000400
HP Virtual LUN Yes SYSTEM
6.0.6001.18000 6/21/2006 Compaq
scsidev.inf Not Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
\5&1BEFC3D4&0&000000
Smart Array P410i Controller No SCSIADAPTER
6.20.0.64 8/24/2009 Hewlett-Packard Company
oem6.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3245103C&REV_0
1\4&2B9E441F&0&0030
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340D&SUBSYS_330B103C&REV_1
3\3&267A616A&0&30
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
20E2D55&0&0400000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
20E2D55&0&0300000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
20E2D55&0&0200000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
20E2D55&0&0100000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&1
20E2D55&0&0000000400000000
Smart Array P411 Controller (Non-Miniport) No
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&18716765&0&0028
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available

```



```

PCI\VEN_8086&DEV_340C&SUBSYS_330B103C&REV_1
3\3&267A616A&0&28
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340B&SUBSYS_330B103C&REV_1
3\3&267A616A&0&20
Intel(R) PRO/1000 PT Quad Port LP Server Adapter Yes
NET 9.11.5.7 6/21/2006 Intel
netele3e.inf Not Available
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&4CC682A&0&01080018
Intel(R) PRO/1000 PT Quad Port LP Server Adapter Yes
NET 9.11.5.7 6/21/2006 Intel
netele3e.inf Not Available
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&4CC682A&0&00080018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_111D&DEV_8018&SUBSYS_00000000&REV_0
4\5&3518D7CE&0&080018
Intel(R) PRO/1000 PT Quad Port LP Server Adapter Yes
NET 9.11.5.7 6/21/2006 Intel
netele3e.inf Not Available
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&21E81E70&0&01000018
Intel(R) PRO/1000 PT Quad Port LP Server Adapter Yes
NET 9.11.5.7 6/21/2006 Intel
netele3e.inf Not Available
PCI\VEN_8086&DEV_10BC&SUBSYS_704B103C&REV_0
6\6&21E81E70&0&00000018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_111D&DEV_8018&SUBSYS_00000000&REV_0
4\5&3518D7CE&0&000018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_111D&DEV_8018&SUBSYS_00000000&REV_0
4\4&5C06900&0&0018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_340A&SUBSYS_330B103C&REV_1
3\3&267A616A&0&18
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3409&SUBSYS_330B103C&REV_1
3\3&267A616A&0&10
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available

```

```

HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
78E328&0&0400000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
78E328&0&0300000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
78E328&0&0200000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
HPQCISS\DISK&VEN_HP&PROD_LOGICAL_VOLUME\5&5
78E328&0&0100000400000000
Smart Array Logical Volume No DISKDRIVE
6.4.0.64 11/7/2008 Hewlett-Packard
oem10.inf Not Available
SCSIADAPTER 6.4.0.64 11/7/2008
Hewlett-Packard oem9.inf Not Available
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_0
1\4&38B514C2&0&0008
PCI standard PCI-to-PCI bridge Yes
SYSTEM 6.0.6002.16670 6/21/2006
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3408&SUBSYS_330B103C&REV_1
3\3&267A616A&0&08
PCI standard host CPU bridge Yes SYSTEM
6.0.6002.16670 6/21/2006 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3406&SUBSYS_330B103C&REV_1
3\3&267A616A&0&00
PCI bus Yes SYSTEM 6.0.6002.16670
6/21/2006 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\0
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\14
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\15
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\14
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\13
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\12

```

```

Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\11
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\10
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\9
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\8
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\7
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\6
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\5
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\4
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\3
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\2
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\1
Intel Processor Yes PROCESSOR
6.0.6001.18000 6/21/2006 Intel
cpu.inf Not Available
ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_26\0
Microsoft ACPI-Compliant System Yes
SYSTEM 6.0.6002.16670 6/21/2006

```



```

11/3/2009 8:38 AM 6.0.6001.18000
27.00 KB (27,648 bytes) 1/19/2008
12:02 AM
taskeng.exe c:\windows\system32\taskeng.exe
1280 6 200 1380
11/3/2009 8:38 AM 6.0.6002.16670
259.00 KB (265,216 bytes) 10/28/2009
10:30 AM
svchost.exe c:\windows\system32\svchost.exe
1492 8 200 1380
11/3/2009 8:38 AM 6.0.6001.18000
27.00 KB (27,648 bytes) 1/19/2008
12:02 AM
svchost.exe c:\windows\system32\svchost.exe
1504 8 200 1380
11/3/2009 8:38 AM 6.0.6001.18000
27.00 KB (27,648 bytes) 1/19/2008
12:02 AM
svchost.exe c:\windows\system32\svchost.exe
1608 8 200 1380
11/3/2009 8:38 AM 6.0.6001.18000
27.00 KB (27,648 bytes) 1/19/2008
12:02 AM
taskeng.exe c:\windows\system32\taskeng.exe
2044 8 200 1380
11/3/2009 8:38 AM 6.0.6002.16670
259.00 KB (265,216 bytes) 10/28/2009
10:30 AM
dwm.exe c:\windows\system32\dwm.exe 856 8
200 1380 11/3/2009 8:38 AM
6.0.6002.16670 96.00 KB (98,304 bytes)
10/28/2009 10:31 AM
explorer.exe c:\windows\explorer.exe
1592 8 200 1380
11/3/2009 8:38 AM 6.0.6002.16670
2.94 MB (3,081,216 bytes) 10/28/2009
10:31 AM
wmiprvse.exe
c:\windows\system32\wbem\wmiprvse.exe
2104 8 200 1380
11/3/2009 8:38 AM 6.0.6002.16670
343.50 KB (351,744 bytes) 10/28/2009
10:30 AM
rundll32.exe c:\windows\system32\rundll32.exe
2272 8 200 1380
11/3/2009 8:38 AM 6.0.6000.16386
45.50 KB (46,592 bytes) 1/19/2008
12:24 AM
rundll32.exe c:\windows\system32\rundll32.exe
2336 8 200 1380
11/3/2009 8:38 AM 6.0.6000.16386
45.50 KB (46,592 bytes) 1/19/2008
12:24 AM
mmc.exe c:\windows\system32\mmc.exe 2452 8
200 1380 11/3/2009 8:38 AM
6.0.6002.16670 2.59 MB (2,715,648
bytes) 10/28/2009 10:31 AM
csrss.exe c:\windows\system32\csrss.exe 988 13
200 1380 11/3/2009 8:40 AM
6.0.6001.18000 7.50 KB (7,680 bytes)
1/18/2008 11:59 PM

```

```

winlogon.exe c:\windows\system32\winlogon.exe
2288 13 200 1380
11/3/2009 8:40 AM 6.0.6002.16670
396.00 KB (405,504 bytes) 10/28/2009
10:30 AM
taskeng.exe c:\windows\system32\taskeng.exe
1680 8 200 1380
11/3/2009 8:40 AM 6.0.6002.16670
259.00 KB (265,216 bytes) 10/28/2009
10:30 AM
rdpclip.exe c:\windows\system32\rdpclip.exe
2680 8 200 1380
11/3/2009 8:40 AM 6.0.6002.16670
187.00 KB (191,488 bytes) 10/28/2009
10:31 AM
dwm.exe c:\windows\system32\dwm.exe 2060 8
200 1380 11/3/2009 8:40 AM
6.0.6002.16670 96.00 KB (98,304 bytes)
10/28/2009 10:31 AM
explorer.exe c:\windows\explorer.exe
2784 8 200 1380
11/3/2009 8:40 AM 6.0.6002.16670
2.94 MB (3,081,216 bytes) 10/28/2009
10:31 AM
msdtc.exe c:\windows\system32\msdtc.exe 2796 8
200 1380 11/3/2009 8:40 AM
2001.12.6931.18000 104.00 KB (106,496
bytes) 1/19/2008 12:27 AM
cmd.exe c:\windows\system32\cmd.exe 2492 8
200 1380 11/3/2009 8:40 AM
6.0.6001.18000 354.50 KB (363,008
bytes) 1/19/2008 12:05 AM
cmd.exe c:\windows\system32\cmd.exe 1164 8
200 1380 11/3/2009 8:53 AM
6.0.6001.18000 354.50 KB (363,008
bytes) 1/19/2008 12:05 AM
sqlservr.exe c:\program files\microsoft sql
server\mssql.1\mssql\binn\sqlservr.exe 2832 13
200 1380 11/3/2009 8:53 AM
2005.90.3042.0 36.72 MB (38,507,376
bytes) 2/10/2007 10:03 AM
osql.exe c:\program files\microsoft sql
server\90\tools\binn\osql.exe 2412 8 200
1380 11/3/2009 8:54 AM
2005.90.3042.0 83.86 KB (85,872 bytes)
2/10/2007 10:02 AM
msinfo32.exe c:\windows\syswow64\msinfo32.exe
2980 8 200 1380
11/3/2009 11:53 AM 6.0.6002.16670
398.50 KB (408,064 bytes) 10/28/2009
10:31 AM
wmiprvse.exe
c:\windows\system32\wbem\wmiprvse.exe 296
8 200 1380 11/3/2009
11:53 AM 6.0.6002.16670 343.50 KB (351,744
bytes) 10/28/2009 10:30 AM
wmiprvse.exe
c:\windows\system32\wbem\wmiprvse.exe
3032 8 200 1380
11/3/2009 11:53 AM 6.0.6002.16670
343.50 KB (351,744 bytes) 10/28/2009
10:30 AM

```

```

taskeng.exe c:\windows\system32\taskeng.exe
2984 6 200 1380
11/3/2009 11:53 AM 6.0.6002.16670
259.00 KB (265,216 bytes) 10/28/2009
10:30 AM
slui.exe c:\windows\system32\slui.exe 2416 8
200 1380 11/3/2009 11:53 AM
6.0.6002.16670 376.00 KB (385,024
bytes) 10/28/2009 10:30 AM
wmiprvse.exe
c:\windows\syswow64\wbem\wmiprvse.exe
1840 8 200 1380
11/3/2009 11:54 AM 6.0.6002.16670
242.00 KB (247,808 bytes) 10/28/2009
10:30 AM
[Loaded Modules]
Name Version Size File Date Manufacturer
Path
csrss 6.0.6001.18000 7.50 KB (7,680 bytes)
1/18/2008 11:59 PM Microsoft Corporation
c:\windows\system32\csrss.exe
ntdll 6.0.6002.16670 1.52 MB (1,598,736
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\ntdll.dll
csrssrv 6.0.6001.18000 83.50 KB (85,504 bytes)
1/18/2008 11:59 PM Microsoft Corporation
c:\windows\system32\csrssrv.dll
basesrv 6.0.6001.18000 78.50 KB (80,384 bytes)
1/18/2008 11:59 PM Microsoft Corporation
c:\windows\system32\basesrv.dll
winsrv 6.0.6002.16670 439.50 KB (450,048
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\winsrv.dll
user32 6.0.6002.16670 801.00 KB (820,224
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\user32.dll
kernel32 6.0.6002.16670 1.16 MB (1,219,072
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\kernel32.dll
gdi32 6.0.6002.16670 380.50 KB (389,632
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\gdi32.dll
advapi32 6.0.6002.16670 1.02 MB (1,065,984
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\advapi32.dll
rpcrt4 6.0.6002.16670 1.25 MB (1,310,208
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\rpcrt4.dll
lpk 6.0.6001.18000 32.00 KB (32,768 bytes)
1/19/2008 12:08 AM Microsoft Corporation
c:\windows\system32\lpk.dll
usp10 1.626.6002.16670 607.00 KB (621,568
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\usp10.dll
msvcrt 7.0.6002.16670 607.50 KB (622,080
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\msvcrt.dll
sxs 6.0.6001.18000 560.50 KB (573,952
bytes) 1/19/2008 12:00 AM Microsoft Corporation
c:\windows\system32\sxs.dll

```

wininit 6.0.6001.18000 121.00 KB (123,904 bytes) 1/19/2008 12:17 AM Microsoft Corporation c:\windows\system32\wininit.exe

userenv 6.0.6001.18000 134.00 KB (137,216 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\userenv.dll

secur32 6.0.6002.16670 93.00 KB (95,232 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\secur32.dll

imm32 6.0.6002.16670 160.50 KB (164,352 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\imm32.dll

msctf 6.0.6002.16670 1,016.50 KB (1,040,896 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\msctf.dll

apphelp 6.0.6002.16670 199.00 KB (203,776 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\apphelp.dll

ws2_32 6.0.6002.16670 258.50 KB (264,704 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\ws2_32.dll

nsi 6.0.6001.18000 11.00 KB (11,264 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\nsi.dll

mswsock 6.0.6002.16670 297.00 KB (304,128 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mswsock.dll

whstcpip 6.0.6001.18000 12.50 KB (12,800 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\whstcpip.dll

wship6 6.0.6001.18000 11.00 KB (11,264 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\wship6.dll

credssp 6.0.6001.18000 18.00 KB (18,432 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\credssp.dll

crypt32 6.0.6002.16670 1.20 MB (1,258,496 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\crypt32.dll

msasn1 6.0.6001.18000 79.00 KB (80,896 bytes) 1/19/2008 12:57 AM Microsoft Corporation c:\windows\system32\msasn1.dll

schannel 6.0.6002.16670 326.50 KB (334,336 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\schannel.dll

netapi32 6.0.6002.16670 634.00 KB (649,216 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\netapi32.dll

psapi 6.0.6001.18000 16.50 KB (16,896 bytes) 1/19/2008 12:40 AM Microsoft Corporation c:\windows\system32\psapi.dll

services 6.0.6002.16670 375.50 KB (384,512 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\services.exe

scserv 6.0.6002.16670 390.00 KB (399,360 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\scserv.dll

authz 6.0.6001.18000 139.50 KB (142,848 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\authz.dll

ncobjapi 6.0.6001.18000 68.50 KB (70,144 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll

ntmarta 6.0.6002.16670 156.00 KB (159,744 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ntmarta.dll

wldap32 6.0.6002.16670 321.00 KB (328,704 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wldap32.dll

samlib 6.0.6002.16670 97.00 KB (99,328 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\samlib.dll

ole32 6.0.6002.16670 1.83 MB (1,915,904 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ole32.dll

lsass 6.0.6001.18000 11.00 KB (11,264 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\lsass.exe

lsasrv 6.0.6002.16670 1.61 MB (1,687,040 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\lsasrv.dll

samsrv 6.0.6002.16670 656.50 KB (672,256 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\samsrv.dll

cryptdll 6.0.6001.18000 63.50 KB (65,024 bytes) 1/19/2008 12:15 AM Microsoft Corporation c:\windows\system32\cryptdll.dll

dnsapi 6.0.6002.16670 216.50 KB (221,696 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\dnsapi.dll

ntdsapi 6.0.6001.18000 143.00 KB (146,432 bytes) 1/19/2008 12:20 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll

feclient 6.0.6002.16670 67.00 KB (68,608 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\feclient.dll

mpr 6.0.6002.16670 83.00 KB (84,992 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mpr.dll

slc 6.0.6002.16670 148.00 KB (151,552 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\slc.dll

sysntfy 6.0.6000.16386 21.00 KB (21,504 bytes) 1/19/2008 12:17 AM Microsoft Corporation c:\windows\system32\sysntfy.dll

wevtapi 6.0.6002.16670 385.00 KB (394,240 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wevtapi.dll

iphlpapi 6.0.6002.16670 124.00 KB (126,976 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll

dhcpcsvc 6.0.6002.16670 262.00 KB (268,288 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\dhcpcsvc.dll

winnsi 6.0.6001.18000 21.50 KB (22,016 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\winnsi.dll

dhcpcsvc6 6.0.6002.16670 160.00 KB (163,840 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\dhcpcsvc6.dll

cngaudit 6.0.6000.16386 14.50 KB (14,848 bytes) 1/19/2008 12:15 AM Microsoft Corporation c:\windows\system32\cngaudit.dll

ncrypt 6.0.6002.16670 247.50 KB (253,440 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ncrypt.dll

bcrypt 6.0.6002.16670 299.50 KB (306,688 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\bcrypt.dll

msprvs 6.0.6000.16386 2.00 KB (2,048 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\msprvs.dll

kerberos 6.0.6002.16670 639.50 KB (654,848 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\kerberos.dll

msvl_0 6.0.6002.16670 259.50 KB (265,728 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\msvl_0.dll

netlogon 6.0.6002.16670 700.00 KB (716,800 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\netlogon.dll

winbrand 6.0.6001.18000 851.00 KB (871,424 bytes) 1/19/2008 12:02 AM Microsoft Corporation c:\windows\system32\winbrand.dll

wdigest 6.0.6001.18000 193.00 KB (197,632 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\wdigest.dll

rsaenh 6.0.6002.16670 282.96 KB (289,752 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rsaenh.dll

tspkg 6.0.6001.18000 77.00 KB (78,848 bytes) 1/19/2008 12:16 AM Microsoft Corporation c:\windows\system32\tspkg.dll

gpapi 6.0.6002.16670 82.50 KB (84,480 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\gpapi.dll

setupapi 6.0.6002.16670 1.83 MB (1,923,584 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\setupapi.dll

oleaut32 6.0.6002.16670 825.50 KB (845,312 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\oleaut32.dll

rassfm 6.0.6001.18000 25.50 KB (26,112 bytes) 1/19/2008 7:51 AM Microsoft Corporation c:\windows\system32\rassfm.dll

scecli 6.0.6002.16670 230.00 KB (235,520 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\scecli.dll

dssenh 6.0.6001.18000 197.55 KB (202,296 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\dssenh.dll

winsta 6.0.6001.18000 200.50 KB (205,312 bytes) 1/19/2008 12:43 AM Microsoft Corporation c:\windows\system32\winsta.dll

cryptnet 6.0.6001.18000 127.00 KB (130,048 bytes) 1/19/2008 12:15 AM Microsoft Corporation c:\windows\system32\cryptnet.dll

sensapi 6.0.6001.18000 12.50 KB (12,800 bytes) 1/19/2008 12:27 AM Microsoft Corporation c:\windows\system32\sensapi.dll

shlwapi 6.0.6002.16670 445.50 KB (456,192 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\shlwapi.dll

comctl32 6.10.6002.16670 1.96 MB (2,050,048 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\winsxs\amd64_microsoft.windows.c

ommon-controls_6595b64144ccf1df_6.0.6002.16670_none_15116808f4081dae\comctl32.dll

lsm 6.0.6001.18000 258.50 KB (264,704 bytes) 1/19/2008 12:43 AM Microsoft Corporation c:\windows\system32\lsm.exe

wmsgapi 6.0.6000.16386 14.00 KB (14,336 bytes) 1/19/2008 12:17 AM Microsoft Corporation c:\windows\system32\wmsgapi.dll

clbcatq 2001.12.6931.18000 597.00 KB (611,328 bytes) 1/19/2008 12:28 AM Microsoft Corporation c:\windows\system32\clbcatq.dll

lsmproxy 6.0.6001.18000 43.50 KB (44,544 bytes) 1/19/2008 12:42 AM Microsoft Corporation c:\windows\system32\lsmproxy.dll

winlogon 6.0.6002.16670 396.00 KB (405,504 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\winlogon.exe

shsvcs 6.0.6002.16670 294.50 KB (301,568 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\shsvcs.dll

svchost 6.0.6001.18000 27.00 KB (27,648 bytes) 1/19/2008 12:02 AM Microsoft Corporation c:\windows\system32\svchost.exe

umpnmgmr 6.0.6002.16670 306.00 KB (313,344 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\umpnmgmr.dll

powrprof 6.0.6002.16670 120.50 KB (123,392 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\powrprof.dll

rpcss 6.0.6002.16670 703.00 KB (719,872 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rpcss.dll

FirewallAPI 6.0.6001.18000 685.00 KB (701,440 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\firewallapi.dll

version 6.0.6002.16670 26.50 KB (27,136 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\version.dll

wtsapi32 6.0.6001.18000 30.50 KB (31,232 bytes) 1/19/2008 12:42 AM Microsoft Corporation c:\windows\system32\wtsapi32.dll

cabinet 6.0.6001.18000 91.00 KB (93,184 bytes) 1/19/2008 12:23 AM Microsoft Corporation c:\windows\system32\cabinet.dll

wintrust 6.0.6001.18000 213.00 KB (218,112 bytes) 1/19/2008 12:15 AM Microsoft Corporation c:\windows\system32\wintrust.dll

imagehlp 6.0.6001.18000 72.50 KB (74,240 bytes) 1/19/2008 12:40 AM Microsoft Corporation c:\windows\system32\imagehlp.dll

fwpuclnt 6.0.6002.16670 761.50 KB (779,776 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\fwpuclnt.dll

wevtvsc 6.0.6002.16670 1.42 MB (1,491,968 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wevtvsc.dll

gpsvc 6.0.6002.16670 702.50 KB (719,360 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\gpsvc.dll

nlaapi 6.0.6001.18000 60.00 KB (61,440 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\nlaapi.dll

wbemprox 6.0.6002.16670 42.50 KB (43,520 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wbemprox.dll

wbemcomn 6.0.6001.18000 516.00 KB (528,384 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wbemcomn.dll

wbemsvc 6.0.6002.16670 121.00 KB (123,904 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wbemsvc.dll

fastprox 6.0.6002.16670 870.50 KB (891,392 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll

profsvc 6.0.6002.16670 174.00 KB (178,176 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\profsvc.dll

atll 3.5.2284.0 85.50 KB (87,552 bytes) 1/19/2008 1:09 AM Microsoft Corporation c:\windows\system32\atll.dll

sens 6.0.6001.18000 60.50 KB (61,952 bytes) 1/19/2008 12:27 AM Microsoft Corporation c:\windows\system32\sens.dll

schedsvcs 6.0.6002.16670 825.00 KB (844,800 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\schedsvcs.dll

ktmw32 6.0.6001.18000 14.50 KB (14,848 bytes) 1/18/2008 11:52 PM Microsoft Corporation c:\windows\system32\ktmw32.dll

comctl32 5.82.6001.18000 619.00 KB (633,856 bytes) 1/19/2008 3:48 AM Microsoft Corporation c:\windows\winsxs\amd64_microsoft.windows.c

ommon-controls_6595b64144ccf1df_5.82.6001.18000_none_40ba501d3c2b20ff\comctl32.dll

taskcomp 6.0.6002.16670 400.00 KB (409,600 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\taskcomp.dll

TSChannel 6.0.6000.16386 18.50 KB (18,944 bytes) 1/19/2008 12:12 AM Microsoft Corporation c:\windows\system32\tschannel.dll

aelupsvc 6.0.6000.16386 26.00 KB (26,624 bytes) 1/18/2008 11:52 PM Microsoft Corporation c:\windows\system32\aelupsvc.dll

ikeext 6.0.6002.16670 444.00 KB (454,656 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ikeext.dll

seclogon 6.0.6001.18000 28.00 KB (28,672 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\seclogon.dll

wmivsc 6.0.6002.16670 216.50 KB (221,696 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wmivsc.dll

srvsvc 6.0.6002.16670 172.50 KB (176,640 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\srvsvc.dll

iphlpvc 6.0.6002.16670 218.50 KB (223,744 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\iphlpvc.dll

rtutils 6.0.6002.16670 49.50 KB (50,688 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rtutils.dll

sqmapi 6.0.6002.16670 171.50 KB (175,616 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\sqmapi.dll

sscore 6.0.6000.16386 12.00 KB (12,288 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\sscore.dll

clusapi 6.0.6001.18000 237.50 KB (243,200 bytes) 1/19/2008 12:05 AM Microsoft Corporation c:\windows\system32\clusapi.dll

activeds 6.0.6001.18000 259.50 KB (265,728 bytes) 1/19/2008 12:19 AM Microsoft Corporation c:\windows\system32\activeds.dll

adslpdc 6.0.6002.16670 225.50 KB (230,912 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\adslpdc.dll

credui 6.0.6001.18000 186.50 KB (190,976 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\credui.dll

shell32 6.0.6002.16670 12.30 MB (12,896,256 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\shell32.dll

resutils 6.0.6001.18000 76.00 KB (77,824 bytes) 1/19/2008 12:04 AM Microsoft Corporation c:\windows\system32\resutils.dll

vssapi 6.0.6002.16670 1.43 MB (1,495,040 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\vssapi.dll

vsstrace 6.0.6001.18000 90.00 KB (92,160 bytes) 1/19/2008 12:29 AM Microsoft Corporation c:\windows\system32\vsstrace.dll

xmllite 1.2.1009.0 176.00 KB (180,224 bytes) 1/19/2008 1:13 AM Microsoft Corporation c:\windows\system32\xmllite.dll

propsys 7.0.6002.16670 901.50 KB (923,136 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\propsys.dll

wbemcore 6.0.6002.16670 1.12 MB (1,173,504 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll

esscli 6.0.6001.18000 418.00 KB (428,032 bytes) 1/19/2008 12:12 AM Microsoft Corporation c:\windows\system32\wbem\esscli.dll

wmiutils 6.0.6001.18000 128.50 KB (131,584 bytes) 1/19/2008 12:12 AM Microsoft Corporation c:\windows\system32\wbem\wmiutils.dll

repdrvfs 6.0.6001.18000 372.50 KB (381,440 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wbem\repdrvfs.dll

wmiprvsd 6.0.6002.16670 687.00 KB (703,488 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll

wbemess 6.0.6001.18000 501.00 KB (513,024 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll

napinsp 6.0.6001.18000 61.50 KB (62,976 bytes) 1/19/2008 12:37 AM Microsoft Corporation c:\windows\system32\napinsp.dll

winrnr 6.0.6001.18000 27.00 KB (27,648 bytes) 1/19/2008 12:19 AM Microsoft Corporation c:\windows\system32\winrnr.dll

ncprov 6.0.6001.18000 77.50 KB (79,360 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll

rasadhlp 6.0.6001.18000 13.00 KB (13,312 bytes) 1/19/2008 12:37 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll

certprop 6.0.6002.16670 48.50 KB (49,664 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\certprop.dll

winscard 6.0.6002.16670 185.50 KB (189,952 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\winscard.dll

sessenv 6.0.6001.18000 73.00 KB (74,752 bytes) 1/19/2008 12:43 AM Microsoft Corporation c:\windows\system32\sessenv.dll

qmgr 7.0.6002.16670 1.03 MB (1,081,344 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\qmgr.dll

shfolder 6.0.6001.18000 10.00 KB (10,240 bytes) 1/19/2008 12:22 AM Microsoft Corporation c:\windows\system32\shfolder.dll

winhttp 6.0.6002.16670 429.50 KB (439,808 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\winhttp.dll

bitsperf 7.0.6000.16386 22.50 KB (23,040 bytes) 1/19/2008 12:11 AM Microsoft Corporation c:\windows\system32\bitsperf.dll

bitsigd 7.0.6001.18000 45.50 KB (46,592 bytes) 1/19/2008 12:11 AM Microsoft Corporation c:\windows\system32\bitsigd.dll

slsvc 6.0.6002.16670 2.46 MB (2,582,016 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\slsvc.exe

es 2001.12.6932.16670 353.00 KB (361,472 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\es.dll

nsisvc 6.0.6001.18000 24.00 KB (24,576 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\nsisvc.dll

wkssvc 6.0.6001.18000 198.00 KB (202,752 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\wkssvc.dll

w32time 6.0.6002.16670 364.00 KB (372,736 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\w32time.dll

netprofm 6.0.6001.18000 297.00 KB (304,128 bytes) 1/19/2008 12:38 AM Microsoft Corporation c:\windows\system32\netprofm.dll

npmproxy 6.0.6000.16386 31.50 KB (32,256 bytes) 1/19/2008 12:38 AM Microsoft Corporation c:\windows\system32\npmproxy.dll

sluinothify 6.0.6002.16670 71.50 KB (73,216 bytes) 10/28/2009 10:30 AM Microsoft Corporation

Corporation c:\windows\system32\sluinothify.dll

slcext 6.0.6002.16670 693.50 KB (710,144 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\slcext.dll

urlmon 7.0.6002.16670 1.35 MB (1,418,752 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\urlmon.dll

iertutil 7.0.6002.16670 366.50 KB (375,296 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\iertutil.dll

uxsms 6.0.6001.18000 32.00 KB (32,768 bytes) 1/19/2008 12:10 AM Microsoft Corporation c:\windows\system32\uxsms.dll

trkwks 6.0.6001.18000 114.50 KB (117,248 bytes) 1/19/2008 12:27 AM Microsoft Corporation c:\windows\system32\trkwks.dll

umrdp 6.0.6002.16670 247.00 KB (252,928 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\umrdp.dll

winspool 6.0.6002.16670 334.00 KB (342,016 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\winspool.drv

umb 6.0.6001.18000 58.50 KB (59,904 bytes) 1/19/2008 12:06 AM Microsoft Corporation c:\windows\system32\umb.dll

netman 6.0.6001.18000 340.00 KB (348,160 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\netman.dll

rasapi32 6.0.6002.16670 329.50 KB (337,408 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rasapi32.dll

rasman 6.0.6001.18000 90.50 KB (92,672 bytes) 1/19/2008 12:37 AM Microsoft Corporation c:\windows\system32\rasman.dll

tapi32 6.0.6000.16386 238.00 KB (243,712 bytes) 1/19/2008 1:13 AM Microsoft Corporation c:\windows\system32\tapi32.dll

winmm 6.0.6002.16670 207.00 KB (211,968 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\winmm.dll

oleacc 4.2.5406.0 300.50 KB (307,712 bytes) 1/19/2008 12:08 AM Microsoft Corporation c:\windows\system32\oleacc.dll

netshell 6.0.6002.16670 3.19 MB (3,341,312 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\netshell.dll

rasdlg 6.0.6002.16670 890.50 KB (911,872 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rasdlg.dll

mprapi 6.0.6002.16670 126.50 KB (129,536 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mprapi.dll

hnetcfg 6.0.6001.18000 423.50 KB (433,664 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll

netcfgx 6.0.6001.18000 492.00 KB (503,808 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\netcfgx.dll

wdi 6.0.6001.18000 80.00 KB (81,920 bytes) 1/19/2008 12:03 AM Microsoft Corporation c:\windows\system32\wdi.dll

radardt 6.0.6000.16386 77.50 KB (79,360 bytes) 1/19/2008 7:52 AM Microsoft Corporation c:\windows\system32\radardt.dll

dnssrslvr 6.0.6002.16670 115.00 KB (117,760 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\dnssrslvr.dll

cryptsv 6.0.6002.16670 162.00 KB (165,888 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\cryptsv.dll

nlavc 6.0.6001.18000 201.50 KB (206,336 bytes) 1/19/2008 12:36 AM Microsoft Corporation c:\windows\system32\nlavc.dll

ncsi 6.0.6001.18000 106.50 KB (109,056 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\ncsi.dll

cfgmgr32 6.0.6001.18000 17.50 KB (17,920 bytes) 1/18/2008 11:59 PM Microsoft Corporation c:\windows\system32\cfgmgr32.dll

esent 6.0.6002.16670 2.39 MB (2,506,752 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\esent.dll

ssdpapi 6.0.6000.16386 49.00 KB (50,176 bytes) 1/19/2008 12:38 AM Microsoft Corporation c:\windows\system32\ssdpapi.dll

termsrv 6.0.6002.16670 534.50 KB (547,328 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\termsrv.dll

icaapi 6.0.6000.16386 20.00 KB (20,480 bytes) 1/19/2008 12:42 AM Microsoft Corporation c:\windows\system32\icaapi.dll

regapi 6.0.6002.16670 87.00 KB (89,088 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\regapi.dll

rdpwsx 6.0.6002.16670 115.00 KB (117,760 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\rdpwsx.dll

mstlsapi 6.0.6002.16670 136.00 KB (139,264 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mstlsapi.dll

msdtckrm 2001.12.6931.18000 386.00 KB (395,264 bytes) 1/19/2008 12:27 AM Microsoft Corporation c:\windows\system32\msdtckrm.dll

wsmvsc 6.0.6002.16670 1.04 MB (1,092,096 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wsmvsc.dll

wsmprov 6.0.6001.18000 71.50 KB (73,216 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wsmprov.dll

winrmgr 6.0.6001.18000 294.00 KB (301,056 bytes) 1/19/2008 12:14 AM Microsoft Corporation c:\windows\system32\winrmgr.dll

httpapi 6.0.6001.18000 32.50 KB (33,280 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\httpapi.dll

wsmres 6.0.6001.18000 13.00 KB (13,312 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\wsmres.dll

wevtfdw 6.0.6001.18000 104.50 KB (107,008 bytes) 1/19/2008 12:12 AM Microsoft Corporation c:\windows\system32\wevtfdw.dll

bfe 6.0.6002.16670 447.50 KB (458,240 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\bfe.dll

mpssvc 6.0.6002.16670 589.00 KB (603,136 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mpssvc.dll

wfapigpp 6.0.6001.18000 20.00 KB (20,480 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\wfapigpp.dll

dps 6.0.6001.18000 136.00 KB (139,264 bytes) 1/19/2008 12:03 AM Microsoft Corporation c:\windows\system32\dps.dll

taskschd 6.0.6001.18000 640.50 KB (655,872 bytes) 1/19/2008 12:13 AM Microsoft Corporation c:\windows\system32\taskschd.dll

taskeng 6.0.6002.16670 259.00 KB (265,216 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\taskeng.exe

dimsjob 6.0.6001.18000 43.00 KB (44,032 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\dimsjob.dll

pautoenr 6.0.6000.16386 46.00 KB (47,104 bytes) 1/19/2008 12:18 AM Microsoft Corporation c:\windows\system32\pautoenr.dll

certcli 6.0.6002.16670 437.00 KB (447,488 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\certcli.dll

certenroll 6.0.6002.16670 1.58 MB (1,658,368 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\certenroll.dll

wininet 7.0.6002.16670 990.50 KB (1,014,272 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wininet.dll

normaliz 6.0.6000.16386 3.00 KB (3,072 bytes) 1/18/2008 11:59 PM Microsoft Corporation c:\windows\system32\normaliz.dll

ipsecsvc 6.0.6002.16670 519.00 KB (531,456 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ipsecsvc.dll

fwremotesvr 6.0.6002.16670 49.00 KB (50,176 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\fwremotesvr.dll

regsvc 6.0.6002.16670 202.00 KB (206,848 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\regsvc.dll

wersvc 6.0.6001.18000 118.00 KB (120,832 bytes) 1/19/2008 12:11 AM Microsoft Corporation c:\windows\system32\wersvc.dll

msctfmonitor 6.0.6002.16670 25.50 KB (26,112 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\msctfmonitor.dll

msutb 6.0.6002.16670 222.50 KB (227,840 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\msutb.dll

dwmapi 6.0.6001.18000 38.50 KB (39,424 bytes) 1/19/2008 12:10 AM Microsoft Corporation c:\windows\system32\dwmapi.dll

PlaySndSrv 6.0.6000.16386 74.50 KB (76,288 bytes) 1/19/2008 12:43 AM Microsoft Corporation

Corporation c:\windows\system32\playsndsrv.dll

mmdevapi 6.0.6002.16670 198.00 KB (202,752 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\mmdevapi.dll

uxtheme 6.0.6001.18000 310.00 KB (317,440 bytes) 1/19/2008 12:21 AM Microsoft Corporation c:\windows\system32\uxtheme.dll

qagent 6.0.6001.18000 245.00 KB (250,880 bytes) 1/19/2008 12:35 AM Microsoft Corporation c:\windows\system32\qagent.dll

qutil 6.0.6001.18000 97.00 KB (99,328 bytes) 1/19/2008 12:34 AM Microsoft Corporation c:\windows\system32\qutil.dll

dwm 6.0.6002.16670 96.00 KB (98,304 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\dwm.exe

dwmredir 6.0.6001.18000 100.00 KB (102,400 bytes) 1/19/2008 12:10 AM Microsoft Corporation c:\windows\system32\dwmredir.dll

milcore 6.0.6001.18000 2.45 MB (2,570,240 bytes) 1/19/2008 12:12 AM Microsoft Corporation c:\windows\system32\milcore.dll

explorer 6.0.6002.16670 2.94 MB (3,081,216 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\explorer.exe

shdocvw 6.0.6002.16670 1.14 MB (1,196,032 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\shdocvw.dll

gdipplus 5.2.6002.16670 2.09 MB (2,192,384 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\winsxs\amd64_microsoft.windows.gdiplus_6595b64144ccf1df_1.0.6002.16670_none_56aaec09b594bdd5_gdiplus.dll

browseui 6.0.6002.16670 1.57 MB (1,651,200 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\browseui.dll

duser 6.0.6001.18000 244.50 KB (250,368 bytes) 1/19/2008 12:09 AM Microsoft Corporation c:\windows\system32\duser.dll

windowscodecs 6.0.6002.16670 822.00 KB (841,728 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\windowscodecs.dll

ehstorshell 5.2.3790.1830 121.00 KB (123,904 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ehstorshell.dll

iconcodservice 6.0.6000.16386 12.50 KB (12,800 bytes) 1/19/2008 7:51 AM Microsoft Corporation c:\windows\system32\iconcodservice.dll

timedate 6.0.6002.16670 860.50 KB (881,152 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\timedate.cpl

ieframe 7.0.6002.16670 6.68 MB (7,005,696 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\ieframe.dll

ExplorerFrame 6.0.6001.18000 39.00 KB (39,936 bytes) 1/19/2008 12:21 AM Microsoft Corporation

Corporation c:\windows\system32\explorerframe.dll

networkexplorer 6.0.6002.16670 2.14 MB (2,247,168 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\networkexplorer.dll

cscapi 6.0.6002.16670 37.50 KB (38,400 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\cscapi.dll

stobject 6.0.6002.16670 731.00 KB (748,544 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\stobject.dll

batmeter 6.0.6001.18000 727.50 KB (744,960 bytes) 1/19/2008 12:23 AM Microsoft Corporation c:\windows\system32\batmeter.dll

sndvol32 6.0.6002.16670 173.50 KB (177,664 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\sndvol32.dll

pnidui 6.0.6002.16670 1.93 MB (2,024,960 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\pnidui.dll

wlanutil 6.0.6000.16386 10.00 KB (10,240 bytes) 1/19/2008 12:34 AM Microsoft Corporation c:\windows\system32\wlanutil.dll

cscui 6.0.6002.16670 657.50 KB (673,280 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\cscui.dll

cscdll 6.0.6002.16670 28.00 KB (28,672 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\cscdll.dll

srchadmin 7.0.6002.16670 339.50 KB (347,648 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\srchadmin.dll

webcheck 7.0.6002.16670 284.00 KB (290,816 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\webcheck.dll

bthprops 6.0.6002.16670 652.00 KB (667,648 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\bthprops.cpl

drprov 6.0.6001.18000 23.50 KB (24,064 bytes) 1/19/2008 12:43 AM Microsoft Corporation c:\windows\system32\drprov.dll

ntlanman 6.0.6001.18000 116.00 KB (118,784 bytes) 1/19/2008 12:14 AM Microsoft Corporation c:\windows\system32\ntlanman.dll

wmiprvse 6.0.6002.16670 343.50 KB (351,744 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wmiprvse.exe

wmi 6.0.6002.16670 223.00 KB (228,352 bytes) 10/28/2009 10:30 AM Microsoft Corporation c:\windows\system32\wbem\wmi.dll

wmi 6.0.6001.18000 5.50 KB (5,632 bytes) 1/19/2008 1:13 AM Microsoft Corporation c:\windows\system32\wmi.dll

rundll32 6.0.6000.16386 45.50 KB (46,592 bytes) 1/19/2008 12:24 AM Microsoft Corporation c:\windows\system32\rundll32.exe

pnmui 5.2.3668.0 675.50 KB (691,712 bytes) 10/28/2009 10:31 AM Microsoft Corporation c:\windows\system32\pnmui.dll

```

newdev 6.0.5054.0 210.50 KB (215,552
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\newdev.dll
mmc 6.0.6002.16670 2.59 MB (2,715,648
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\mmc.exe
mfc42u 6.6.8063.0 1.29 MB (1,357,824
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\mfc42u.dll
odbc32 6.0.6002.16670 452.00 KB (462,848
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\odbc32.dll
comdlg32 6.0.6002.16670 537.00 KB (549,888
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\comdlg32.dll
mmcbase 6.0.6001.18000 342.00 KB (350,208
bytes) 1/19/2008 12:11 AM Microsoft Corporation
c:\windows\system32\mmcbase.dll
odbcint 6.0.6000.16386 224.00 KB (229,376
bytes) 1/19/2008 12:56 AM Microsoft Corporation
c:\windows\system32\odbcint.dll
mmcndmgr 6.0.6002.16670 3.11 MB (3,263,488
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\mmcndmgr.dll
msxml3 8.100.5000.0 1.72 MB (1,803,264
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\msxml3.dll
mlang 6.0.6001.18000 232.50 KB (238,080
bytes) 1/19/2008 12:22 AM Microsoft Corporation
c:\windows\system32\mlang.dll
mscoree 2.0.50727.3074 394.82 KB (404,296
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\mscoree.dll
mscorwks 2.0.50727.3074 9.61 MB (10,077,512
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\microsoft.net\framework64\v2.0.5
0727\mscorwks.dll
msvcr80 8.0.50727.3074 784.82 KB (803,656
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\winsxs\amd64_microsoft.vc80.crt_
1fc8b3b9a1e18e3b_8.0.50727.3074_none_88e22eaf2facbe7f
\msvcr80.dll
mscorlib.ni 2.0.50727.3074 14.84 MB
(15,558,656 bytes) 10/28/2009 10:56 AM Microsoft
Corporation
c:\windows\assembly\nativeimages_v2.0.50727
_64\mscorlib\ec1a6138e295413aa4134c181cb27da7\mscorli
b.ni.dll
mmcx 6.0.6002.16670 408.00 KB (417,792
bytes) 10/28/2009 10:31 AM Not Available
c:\windows\assembly\gac_msil\mmcx\3.0.0.0_
31bf3856ad364e35\mmcx.dll
mscorjit 2.0.50727.3074 1.50 MB (1,577,800
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\microsoft.net\framework64\v2.0.5
0727\mscorjit.dll
mmcfxcommon 6.0.6002.16670 108.00 KB
(110,592 bytes) 10/28/2009 10:31 AM Not Available
c:\windows\assembly\gac_msil\mmcfxcommon\3.
0.0.0_31bf3856ad364e35\mmcfxcommon.dll
System.ni 2.0.50727.3074 10.00 MB (10,486,272
bytes) 10/28/2009 10:56 AM Microsoft Corporation

```

```

c:\windows\assembly\nativeimages_v2.0.50727
_64\system\c2c1a8a88ada33e9f80e59ab008b7f19\system.ni
.dll
system.configuration 2.0.50727.3074
416.00 KB (425,984 bytes) 10/28/2009
10:31 AM Microsoft Corporation
c:\windows\assembly\gac_msil\system.configu
ration\2.0.0.0_b03f5f7f11d50a3a\system.configuration
.dll
System.Xml.ni 2.0.50727.3074 6.63 MB
(6,948,352 bytes) 10/28/2009 10:57 AM Microsoft
Corporation
c:\windows\assembly\nativeimages_v2.0.50727
_64\system.xml\17f5ca8173b34f2395ab19afb63c76ca\sys
tem.xml.ni.dll
System.Drawing.ni 2.0.50727.3074 2.21 MB
(2,312,704 bytes) 10/28/2009 10:57 AM Microsoft
Corporation
c:\windows\assembly\nativeimages_v2.0.50727
_64\system.drawing\9e091aed0d3e422ef295d01ae1baa87\
system.drawing.ni.dll
System.Windows.Forms.ni 2.0.50727.3074
16.57 MB (17,377,792 bytes) 10/28/2009
10:57 AM Microsoft Corporation
c:\windows\assembly\nativeimages_v2.0.50727
_64\system.windows.forms\d35f83f9f29671a724f478897ccc
2c5e\system.windows.forms.ni.dll
diasymreader 8.0.50727.3074 778.83 KB
(797,520 bytes) 10/28/2009 10:31 AM Microsoft
Corporation
c:\windows\microsoft.net\framework64\v2.0.5
0727\diasymreader.dll
microsoft.managementconsole 6.0.6002.16670
184.00 KB (188,416 bytes) 10/28/2009
10:31 AM Not Available
c:\windows\assembly\gac_msil\microsoft.man
agementconsole\3.0.0.0_31bf3856ad364e35\microsoft.man
agementconsole.dll
microsoft.windows.servermanager
6.0.6002.16670 7.29 MB (7,639,040
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\assembly\gac_msil\microsoft.wind
ows.servermanager\6.0.0.0_31bf3856ad364e35\microsoft
.windows.servermanager.dll
dciman32 6.0.6001.18000 14.00 KB (14,336 bytes)
1/19/2008 12:07 AM Microsoft Corporation
c:\windows\system32\dciman32.dll
miguicontrols 6.0.6002.16670 3.22 MB
(3,375,104 bytes) 10/28/2009 10:31 AM Not Available
c:\windows\assembly\gac_msil\miguicontrols\
1.0.0.0_31bf3856ad364e35\miguicontrols.dll
system.serviceprocess 2.0.50727.3074
112.00 KB (114,688 bytes) 10/28/2009
10:30 AM Microsoft Corporation
c:\windows\assembly\gac_msil\system.servic
eprocess\2.0.0.0_b03f5f7f11d50a3a\system.serviceproce
ss.dll
accessibility 2.0.50727.3074 10.50 KB
(10,752 bytes) 10/28/2009 10:31 AM Microsoft
Corporation
c:\windows\assembly\gac_msil\accessibility\
2.0.0.0_b03f5f7f11d50a3a\accessibility.dll

```

```

system.configuration.install 2.0.50727.3074
80.00 KB (81,920 bytes) 10/28/2009
10:31 AM Microsoft Corporation
c:\windows\assembly\gac_msil\system.configu
ration.install\2.0.0.0_b03f5f7f11d50a3a\system.conf
iguration.install.dll
rmconfighelper 6.0.6002.16670 1.18 MB
(1,236,992 bytes) 10/28/2009 10:31 AM Not Available
c:\windows\assembly\gac_msil\rmconfighelper
\6.0.0.0_31bf3856ad364e35\rmconfighelper.dll
svrMgrnc 6.0.6002.16670 192.00 KB (196,608
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\svrMgrnc.dll
osbaseln 6.0.6001.18000 25.00 KB (25,600 bytes)
1/19/2008 12:07 AM Microsoft Corporation
c:\windows\system32\osbaseln.dll
wuapi 7.0.6002.16670 640.50 KB (655,872
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\wuapi.dll
rdpclip 6.0.6002.16670 187.00 KB (191,488
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\rdpclip.exe
actxprxy 6.0.6001.18000 979.00 KB (1,002,496
bytes) 1/19/2008 1:13 AM Microsoft Corporation
c:\windows\system32\actxprxy.dll
ntshrui 6.0.6001.18000 347.00 KB (355,328
bytes) 1/19/2008 12:24 AM Microsoft Corporation
c:\windows\system32\ntshrui.dll
msdtc 2001.12.6931.18000 104.00 KB (106,496
bytes) 1/19/2008 12:27 AM Microsoft Corporation
c:\windows\system32\msdtc.exe
msdtctm 2001.12.6932.16670 1.43 MB (1,499,136
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\msdtctm.dll
msdtcprx 2001.12.6932.16670 710.50 KB (727,552
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\msdtcprx.dll
mtxclu 2001.12.6932.16670 353.50 KB (361,984
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\mtxclu.dll
msdtclog 2001.12.6931.18000 113.00 KB (115,712
bytes) 1/19/2008 12:27 AM Microsoft Corporation
c:\windows\system32\msdtclog.dll
xolehlp 2001.12.6931.18000 47.00 KB (48,128 bytes)
1/19/2008 12:27 AM Microsoft Corporation
c:\windows\system32\xolehlp.dll
comres 2001.12.6931.18000 1.23 MB (1,291,264
bytes) 1/19/2008 12:27 AM Microsoft Corporation
c:\windows\system32\comres.dll
msdtcVSpres 2001.12.6931.18000 20.50 KB
(20,992 bytes) 1/19/2008 12:27 AM Microsoft
Corporation
c:\windows\system32\msdtcvspres.dll
mtxoci 2001.12.6931.18000 148.00 KB (151,552
bytes) 1/19/2008 12:27 AM Microsoft Corporation
c:\windows\system32\mtxoci.dll
cmd 6.0.6001.18000 354.50 KB (363,008
bytes) 1/19/2008 12:05 AM Microsoft Corporation
c:\windows\system32\cmd.exe
sqlservr 2005.90.3042.0 36.72 MB (38,507,376
bytes) 2/10/2007 10:03 AM Microsoft Corporation

```



```

c:\program files\microsoft sql
server\mssql1.1\mssql\bin\sqlservr.exe
msvcp80 8.0.50727.3074 1.02 MB (1,069,384
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\winsxs\amd64_microsoft_vc80.crt_
1fc8b3b9a1e18e3b_8.0.50727.3074_none_88e22eaf2facbe7f
\msvcp80.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\mssql1.1\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 2005.90.3042.0 1.66 MB (1,740,656
bytes) 2/10/2007 10:02 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql1.1\mssql\bin\resources\1033\sqllevn70.rll
sqlos 2005.90.3042.0 17.86 KB (18,288 bytes)
2/10/2007 10:03 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql1.1\mssql\bin\sqlos.dll
security 6.0.6000.16386 5.50 KB (5,632 bytes)
1/19/2008 12:16 AM Microsoft Corporation
c:\windows\system32\security.dll
msfte 12.0.6828.0 3.63 MB (3,804,952
bytes) 8/28/2006 5:17 AM Microsoft Corporation
c:\program files\microsoft sql
server\mssql1.1\mssql\bin\msfte.dll
dbghelp 6.6.7.5 1.27 MB (1,329,520 bytes)
2/10/2007 9:56 AM Microsoft Corporation
c:\program files\microsoft sql
server\90\shared\dbghelp.dll
sqlncli 2005.90.3042.0 2.74 MB (2,868,592
bytes) 2/10/2007 10:03 AM Microsoft Corporation
c:\windows\system32\sqlncli.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
osql 2005.90.3042.0 83.86 KB (85,872 bytes)
2/10/2007 10:02 AM Microsoft Corporation
c:\program files\microsoft sql
server\90\tools\bin\osql.exe
osql 2005.90.1399.0 15.21 KB (15,576 bytes)
10/14/2005 3:31 PM Microsoft Corporation
c:\program files\microsoft sql
server\90\tools\bin\resources\1033\osql.rll
msinfo32 6.0.6002.16670 398.50 KB (408,064
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\syswow64\msinfo32.exe
wow64 6.0.6002.16670 229.00 KB (234,496
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\wow64.dll
wow64win 6.0.6002.16670 294.50 KB (301,568
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\wow64win.dll
wow64cpu 6.0.6002.16670 17.00 KB (17,408 bytes)
10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\wow64cpu.dll

```

```

cimwin32 6.0.6002.16670 1.99 MB (2,082,816
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\wbem\cimwin32.dll
framedynos 6.0.6001.18000 275.00 KB
(281,600 bytes) 1/19/2008 12:13 AM Microsoft
Corporation
c:\windows\system32\framedynos.dll
ntevt 6.0.6002.16670 251.00 KB (257,024
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\wbem\ntevt.dll
provthrd 6.0.6001.18000 327.50 KB (335,360
bytes) 1/19/2008 12:13 AM Microsoft Corporation
c:\windows\system32\provthrd.dll
msvcirt 7.0.6000.16386 78.50 KB (80,384 bytes)
1/18/2008 11:52 PM Microsoft Corporation
c:\windows\system32\msvcirt.dll
wsock32 6.0.6001.18000 18.00 KB (18,432 bytes)
1/19/2008 12:37 AM Microsoft Corporation
c:\windows\system32\wsock32.dll
slui 6.0.6002.16670 376.00 KB (385,024
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\slui.exe
slcommdlg 6.0.6002.16670 616.50 KB (631,296
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\system32\slcommdlg.dll
msimg32 6.0.6001.18000 8.00 KB (8,192 bytes)
1/19/2008 12:07 AM Microsoft Corporation
c:\windows\system32\msimg32.dll
slwga 6.0.6001.18000 14.00 KB (14,336 bytes)
1/19/2008 12:17 AM Microsoft Corporation
c:\windows\system32\slwga.dll
cryptui 6.0.6002.16670 1,011.50 KB (1,035,776
bytes) 10/28/2009 10:31 AM Microsoft Corporation
c:\windows\system32\cryptui.dll
scarddlg 6.0.6000.16386 81.00 KB (82,944 bytes)
1/19/2008 12:15 AM Microsoft Corporation
c:\windows\system32\scarddlg.dll
wmiprvse 6.0.6002.16670 242.00 KB (247,808
bytes) 10/28/2009 10:30 AM Microsoft Corporation
c:\windows\syswow64\wbem\wmiprvse.exe

```

[Services]

Display Name	Name	State	Start Mode	Path	Error Control
	Service Type				
	Start Name			Tag ID	
Application Experience	AeLookupSvc	Running	Auto	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				
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\framework64\v2.0.5
0727\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\v2.0.507
27\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\framework64\v2.0.5
0727\mscorsvw.exe Ignore LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dlh\host.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
Offline Files CscService Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k
localsystemnetworkrestricted Normal LocalSystem
0
DCOM Server Process Launcher DcomLaunch
Running Auto Share Process

```

```

c:\windows\system32\svchost.exe -k
dcomlaunch Normal LocalSystem 0

DFS Namespace Dfs Stopped Manual Own
Process c:\windows\system32\dfssvc.exe
Normal LocalSystem 0

DHCP Client Dhcp Running Auto
Share Process
c:\windows\system32\svchost.exe -k
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

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

Group Policy Client gpsvc Running Auto Own
Process c:\windows\system32\svchost.exe -k
gpsvcgroup 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

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 Running Auto Share Process
c:\windows\system32\svchost.exe -k
networkservice 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
localservice Normal NT
AUTHORITY\LocalService 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 Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k
localservicenetworkrestricted Normal NT
AUTHORITY\LocalService 0

Multimedia Class Scheduler MMCSS 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

SQL Server FullText Search (MSSQLSERVER)
msftesql Stopped Manual Own Process
"c:\program files\microsoft sql

```

```

server\mssql.1\mssql\bin\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 /v
Normal LocalSystem 0

SQL Server (MSSQLSERVER) MSSQLSERVER
Stopped Manual Own Process
"c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlservr.exe" -smssqlserver
Normal LocalSystem 0

SQL Server Active Directory Helper
MSSQLServerADHelper Stopped Disabled Own
Process "c:\program files\microsoft sql
server\90\shared\sqladhlp90.exe" Normal NT
AUTHORITY\NetworkService 0

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
Auto Share Process
c:\windows\system32\svchost.exe -k
localservice 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

Removable Storage NtmsSvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -k rsmvcs
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
local servicenonetwork     Normal  NT
AUTHORITY\LocalService    0
Plug and Play              PlugPlay Running  Auto
Share Process
c:\windows\system32\svchost.exe -k
dcomlaunch                 Normal  LocalSystem 0

IPsec Policy Agent         PolicyAgent Running
Auto Share Process
c:\windows\system32\svchost.exe -k
networkservicenetworkrestricted Normal  NT
Authority\NetworkService  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 regsvcs
Normal NT AUTHORITY\LocalService 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

Remote Access Quarantine Agent Rqs
Stopped Manual Own Process
c:\windows\system32\rqs.exe Normal NT
AUTHORITY\LocalService 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 Manager SamSs 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
localservice 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 Running Auto
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
Terminal Services 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
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
Software Licensing slsvc Running Auto Own
Process c:\windows\system32\slsvc.exe Normal NT
AUTHORITY\NetworkService 0
SL UI Notification Service SLUINotify
Running Manual Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 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
SQL Server Browser        SQLBrowser Stopped
Disabled Own Process "c:\program
files (x86)\microsoft sql
server\90\shared\sqlbrowser.exe"
LocalSystem 0
SQL Server Agent (MSSQLSERVER)
SQLSERVERAGENT Stopped Manual Own
Process "c:\program files\microsoft sql
server\mssql.1\mssql\bin\sqlagent90.exe" -i

```

```

mssqlserver Normal LocalSystem 0

SQL Server VSS Writer      SQLWriter Stopped
Manual Own Process "c:\program
files\microsoft sql server\90\shared\sqlwriter.exe"
Normal LocalSystem 0
SSDP Discovery            SSDPSRV Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k
localservice 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
Superfetch               SysMain Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k
localsystemnetworkrestricted Ignore 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 Auto
Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Terminal Services TermService Running
Auto Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
Authority\NetworkService 0
Themes Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Thread Ordering Server THREADORDER
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows\system32\tlntsvr.exe
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 Disabled Own Process
c:\windows\servicing\trustedinstaller.exe
Normal localSystem 0
Interactive Services Detection UI0Detect
Stopped Manual Own Process

```

```

c:\windows\system32\ui0detect.exe
Normal LocalSystem 0
Terminal Services UserMode Port Redirector
UmRdpService Running Manual
Share Process
c:\windows\system32\svchost.exe -k
localsystemnetworkrestricted Normal localSystem
0
UPnP Device Host upnphost Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k
localservice 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
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 Auto
Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Windows Color System WcsPlugInService
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k wcsvvc
Normal NT AUTHORITY\LocalService 0

Diagnostic Service Host WdiServiceHost
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k wdisvc
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 Wecsvc Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Problem Reports and Solutions Control Panel Support
werclpsupport Stopped Manual
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal localSystem 0
Windows Error Reporting Service WerSvc
Running Auto Share Process
c:\windows\system32\svchost.exe -k
wersvcgroup Ignore localSystem 0

WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped Manual
Share Process

```

```

c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Windows Management Instrumentation Winmgmt
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore localSystem 0
Windows Remote Management (WS-Management)
WinRM Running Auto 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\wmiapsrv.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 wuauerv Stopped Disabled
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
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\Entertainment
Default:Start
Menu\Programs\Accessories\Entertainment 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\Programs\startup Default:Start
Menu\Programs\startup 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\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\Extras and Upgrades Public:Start
Menu\Programs\Extras and Upgrades 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\HP System Tools\HP Array
Configuration Utility CLI Public:Start
Menu\Programs\HP System Tools\HP Array Configuration
Utility CLI Public
Start Menu\Programs\HP System Tools\HP Array
Diagnostic Utility Public:Start Menu\Programs\HP
System Tools\HP Array Diagnostic 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\Analysis Services Public:Start
Menu\Programs\Microsoft SQL Server 2005\Analysis
Services 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
2005\Documentation and Tutorials Public:Start
Menu\Programs\Microsoft SQL Server 2005\Documentation
and Tutorials Public
Start Menu\Programs\Microsoft SQL Server
2005\Documentation and Tutorials\Tutorials
Public:Start Menu\Programs\Microsoft SQL
Server 2005\Documentation and Tutorials\Tutorials
Public
Start Menu\Programs\Microsoft SQL Server
2005\Performance Tools Public:Start
Menu\Programs\Microsoft SQL Server 2005\Performance
Tools Public
Start Menu\Programs\Microsoft Visual Studio 2005
Public:Start Menu\Programs\Microsoft Visual
Studio 2005 Public
Start Menu\Programs\Microsoft Visual Studio
2005\Visual Studio Tools Public:Start
Menu\Programs\Microsoft Visual Studio 2005\Visual
Studio Tools Public
Start Menu\Programs\PLX SDK v6.20 Public:Start
Menu\Programs\PLX SDK v6.20 Public

```

```

Start Menu\Programs\PLX SDK v6.20\Documentation
      Public:Start Menu\Programs\PLX SDK
v6.20\Documentation Public
Start Menu\Programs\Startup      Public:Start
Menu\Programs\Startup           Public
Start Menu      HOOPS\Administrator:Start Menu
      HOOPS\Administrator
Start Menu\Programs HOOPS\Administrator:Start
Menu\Programs      HOOPS\Administrator
Start Menu\Programs\Accessories
      HOOPS\Administrator:Start
Menu\Programs\Accessories HOOPS\Administrator
Start Menu\Programs\Accessories\Accessibility
      HOOPS\Administrator:Start
Menu\Programs\Accessories\Accessibility
      HOOPS\Administrator
Start Menu\Programs\Accessories\System Tools
      HOOPS\Administrator:Start
Menu\Programs\Accessories\System Tools
      HOOPS\Administrator
Start Menu\Programs\Administrative Tools
      HOOPS\Administrator:Start
Menu\Programs\Administrative Tools
      HOOPS\Administrator
Start Menu\Programs\KrView
      HOOPS\Administrator:Start
Menu\Programs\KrView      HOOPS\Administrator
Start Menu\Programs\Maintenance
      HOOPS\Administrator:Start
Menu\Programs\Maintenance HOOPS\Administrator
Start Menu\Programs\Startup
      HOOPS\Administrator:Start
Menu\Programs\Startup      HOOPS\Administrator

[Startup Programs]

Program Command User Name Location
QLogicSaveSystemInfo rundll32.exe
qlco1006.dll,qlsavesysteminfo Public
ion\Run      HKLM\SOFTWARE\Microsoft\Windows\CurrentVers

[OLE Registration]

Object Local Server
WordPad Document "%programfiles%\windows
nt\accessories\wordpad.exe"
Package Not Available

[Windows Error Reporting]

Time Type Details

```

Server Bus Performance

Driver Registry Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissb
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:01 AM
Value 0
Name: DisplayName
Type: REG_SZ
Data: Smart Array Controllers Non-
Miniport Bus Driver

Value 1
Name: Type
Type: REG_DWORD
Data: 0x1

Value 2
Name: Start
Type: REG_DWORD
Data: 0

Value 3
Name: ErrorControl
Type: REG_DWORD
Data: 0x1

Value 4
Name: ImagePath
Type: REG_EXPAND_SZ
Data: system32\DRIVERS\hpqcissb.sys

Value 5
Name: Group
Type: REG_SZ
Data: port

Value 6
Name: Tag
Type: REG_DWORD
Data: 0x102

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissb\Enum
Class Name: <NO CLASS>
Last Write Time: 10/28/2009 - 11:15 AM
Value 0
Name: Count
Type: REG_DWORD
Data: 0xe

Value 1
Name: NextInstance
Type: REG_DWORD
Data: 0xe

```

```

Value 2
Name: 1
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&11ff4a
0c&0&0048

Value 3
Name: 2
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&187167
65&0&0028

Value 4
Name: 3
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&190f99
ae&0&0040

Value 5
Name: 4
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&294c1e
27&0&00E0

Value 6
Name: 5
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&2b0041
d6&0&0038

Value 7
Name: 6
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&3892f0
ce&0&0008

Value 8
Name: 7
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&38b514
c2&0&0008

Value 9
Name: 8
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&16266d
a6&0&00080038

Value 10
Name: 9
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&179cc5
72&0&00500038

```

```

Value 11
Name: 10
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&2fe19b63&0&00680038

Value 12
Name: 11
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&326db53e&0&00600038

Value 13
Name: 12
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&3b2c24d1&0&00480038

Value 14
Name: 13
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\6&c523c04&0&00700038

Value 15
Name: 0
Type: REG_SZ
Data:
PCI\VEN_103C&DEV_323A&SUBSYS_3247103C&REV_01\4&59e450c&0&0018

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissb\Parameters
Class Name: <NO CLASS>
Last Write Time: 10/29/2009 - 10:53 AM
Value 0
Name: CompletionMode
Type: REG_DWORD
Data: 0x2

Value 1
Name: CosTimerRate
Type: REG_DWORD
Data: 0x2

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissb\Security
Class Name: <NO CLASS>
Last Write Time: 10/27/2009 - 4:49 PM
Value 0
Name: Security
Type: REG_BINARY
Data:

```

```

00000000 01 00 14 80 b8 00 00 00 - c4 00 00 00 14
00 00 00 .....Ä.....
00000010 30 00 00 00 02 00 1c 00 - 01 00 00 00 02
80 14 00 0.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00
00 00 00 ý.....
00000030 02 00 88 00 06 00 00 00 - 00 00 14 00 fd
01 02 00 .....ÿ...
00000040 01 01 00 00 00 00 00 05 - 12 00 00 00 00
00 18 00 .....
00000050 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20
00 00 00 ý.....
00000060 20 02 00 00 00 00 14 00 - 8d 01 02 00 01
01 00 00 .....
00000070 00 00 00 05 04 00 00 00 - 00 00 14 00 8d
01 02 00 .....
00000080 01 01 00 00 00 00 05 05 - 06 00 00 00 00
00 14 00 .....
00000090 00 01 00 00 01 01 00 00 - 00 00 00 05 0b
00 00 00 .....
000000a0 00 00 18 00 fd 01 02 00 - 01 02 00 00 00
00 00 05 ...ÿ.....
000000b0 20 00 00 00 23 02 00 00 - 01 01 00 00 00
00 00 05 ...#.....
000000c0 12 00 00 00 01 01 00 00 - 00 00 00 05 12
00 00 00 .....

```

Server Disk Device Performance Driver Registry Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissd
Class Name: <NO CLASS>
Last Write Time: 11/4/2009 - 3:24 PM
Value 0
Name: Type
Type: REG_DWORD
Data: 0x1

Value 1
Name: Start
Type: REG_DWORD
Data: 0

Value 2
Name: ErrorControl
Type: REG_DWORD

```

```

Data: 0x1

Value 3
Name: Tag
Type: REG_DWORD
Data: 0x102

Value 4
Name: ImagePath
Type: REG_EXPAND_SZ
Data: system32\DRIVERS\hpqcissd.sys

Value 5
Name: DisplayName
Type: REG_SZ
Data: Smart Array Controllers Non-
Miniport Disk Driver

Value 6
Name: Group
Type: REG_SZ
Data: Primary Disk

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
hpqcissd\Enum
Class Name: <NO CLASS>
Last Write Time: 11/4/2009 - 3:24 PM
Value 0
Name: 0
Type: REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&578e328&0&0
0000040000000000

Value 1
Name: Count
Type: REG_DWORD
Data: 0x41

Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x41

Value 3
Name: 1
Type: REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&578e328&0&0
1000040000000000

Value 4
Name: 2
Type: REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&578e328&0&0
2000040000000000

Value 5
Name: 3

```

Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&578e328&0&0300040000000000

Value 6
 Name: 4
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&578e328&0&0400040000000000

Value 7
 Name: 5
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&120e2d55&0&0000040000000000

Value 8
 Name: 6
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&120e2d55&0&0100004000000000

Value 9
 Name: 7
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&120e2d55&0&0200004000000000

Value 10
 Name: 8
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&120e2d55&0&0300004000000000

Value 11
 Name: 9
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&120e2d55&0&0400004000000000

Value 12
 Name: 10
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1657c522&0&0000004000000000

Value 13
 Name: 11
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1657c522&0&0100004000000000

Value 14

Name: 12
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1657c522&0&0200004000000000

Value 15
 Name: 13
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1657c522&0&0300004000000000

Value 16
 Name: 14
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1657c522&0&0400004000000000

Value 17
 Name: 15
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&daeb7ed&0&0000040000000000

Value 18
 Name: 16
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&daeb7ed&0&0100004000000000

Value 19
 Name: 17
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&daeb7ed&0&0200004000000000

Value 20
 Name: 18
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&daeb7ed&0&0300004000000000

Value 21
 Name: 19
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&daeb7ed&0&0400004000000000

Value 22
 Name: 20
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&481c591&0&0000040000000000

Value 23
 Name: 21
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&481c591&0&0100004000000000

Value 24
 Name: 22
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&481c591&0&0200004000000000

Value 25
 Name: 23
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&481c591&0&0300004000000000

Value 26
 Name: 24
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&481c591&0&0400004000000000

Value 27
 Name: 25
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&19711c8e&0&0000004000000000

Value 28
 Name: 26
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&19711c8e&0&0100004000000000

Value 29
 Name: 27
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&19711c8e&0&0200004000000000

Value 30
 Name: 28
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&19711c8e&0&0300004000000000

Value 31
 Name: 29
 Type: REG_SZ
 Data: HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&19711c8e&0&0400004000000000

Value 32
 Name: 30
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&2ced7d2f&0&0000040000000000

Value 33
 Name: 31
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&2ced7d2f&0&0100004000000000

Value 34
 Name: 32
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&2ced7d2f&0&0200004000000000

Value 35
 Name: 33
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&2ced7d2f&0&0300004000000000

Value 36
 Name: 34
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&2ced7d2f&0&0400004000000000

Value 37
 Name: 35
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1477dab3&0&0000004000000000

Value 38
 Name: 36
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1477dab3&0&0100004000000000

Value 39
 Name: 37
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1477dab3&0&0200004000000000

Value 40
 Name: 38
 Type: REG_SZ

Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1477dab3&0&0300004000000000

Value 41
 Name: 39
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\7&1477dab3&0&0400004000000000

Value 42
 Name: 40
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&59169f4&0&0000040000000000

Value 43
 Name: 41
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&59169f4&0&0100004000000000

Value 44
 Name: 42
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&59169f4&0&0200004000000000

Value 45
 Name: 43
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&59169f4&0&0300004000000000

Value 46
 Name: 44
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&59169f4&0&0400004000000000

Value 47
 Name: 45
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&10fb3843&0&0000004000000000

Value 48
 Name: 46
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&10fb3843&0&0100004000000000

Value 49
 Name: 47

Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&10fb3843&0&0200004000000000

Value 50
 Name: 48
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&10fb3843&0&0300004000000000

Value 51
 Name: 49
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&10fb3843&0&0400004000000000

Value 52
 Name: 50
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&8ca42ae&0&0000040000000000

Value 53
 Name: 51
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&8ca42ae&0&0100004000000000

Value 54
 Name: 52
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&8ca42ae&0&0200004000000000

Value 55
 Name: 53
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&8ca42ae&0&0300004000000000

Value 56
 Name: 54
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&8ca42ae&0&0400004000000000

Value 57
 Name: 55
 Type: REG_SZ
 Data:
 HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&340d2586&0&0000004000000000

Value 58


```

Name:          56
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&340d2586&0&
0100004000000000

Value 59
Name:          57
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&340d2586&0&
0200004000000000

Value 60
Name:          58
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&340d2586&0&
0300004000000000

Value 61
Name:          59
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&340d2586&0&
0400004000000000

Value 62
Name:          60
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&2add7011&0&
0000004000000000

Value 63
Name:          61
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&2add7011&0&
0100004000000000

Value 64
Name:          62
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&2add7011&0&
0200004000000000

Value 65
Name:          63
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&2add7011&0&
0300004000000000

Value 66
Name:          64
Type:          REG_SZ
Data:
HPQCISS\Disk&VEN_HP&PROD_LOGICAL_VOLUME\5&2add7011&0&
0400004000000000

```

Web Client Hardware Configuration

System Information report written at: 11/03/09
11:30:50
System Name: CL121
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Standard Edition
Version	5.2.3790 Service Pack 2 Build 3790
Other OS Description	R2
OS Manufacturer	Microsoft Corporation
System Name	CL121
System Manufacturer	HP
System Model	ProLiant DL360 G5
System Type	X86-based PC
Processor x86 Family 6 Model 23 Stepping 6	
GenuineIntel ~2833 Mhz	
Processor x86 Family 6 Model 23 Stepping 6	
GenuineIntel ~2833 Mhz	
Processor x86 Family 6 Model 23 Stepping 6	
GenuineIntel ~2833 Mhz	
BIOS Version/Date	HP P58, 1/24/2008
SMBIOS Version	2.4
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer Version =	"5.2.3790.3959 (srv03_sp2_rtm.070216-1710)"
User Name	Not Available
Time Zone	Central Standard Time
Total Physical Memory	1,021.86 MB
Available Physical Memory	816.57 MB
Total Virtual Memory	2.91 GB
Available Virtual Memory	2.82 GB
Page File Space	2.00 GB
Page File	C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device	Status
I/O Port	0x00000000-0x00000CF7	PCI bus
I/O Port	0x00000000-0x00000CF7	Direct memory access controller

I/O Port	0x00002F8-0x00002FF	Motherboard resources
I/O Port	0x00002F8-0x00002FF	Communications Port (COM2)
IRQ	22	HP iLO Management Channel Interface Driver
IRQ	22	Standard Universal PCI to USB Host Controller
IRQ	16	PCI standard PCI-to-PCI bridge
IRQ	16	Smart Array P400I Controller
IRQ	16	Standard Universal PCI to USB Host Controller
IRQ	16	Standard Enhanced PCI to USB Host Controller
IRQ	17	PCI standard PCI-to-PCI bridge
IRQ	17	Standard Universal PCI to USB Host Controller
IRQ	18	PCI standard PCI-to-PCI bridge
IRQ	18	HP NC373i Virtual Bus Device
IRQ	18	Standard Universal PCI to USB Host Controller
IRQ	19	HP NC373i Virtual Bus Device
IRQ	19	Standard Universal PCI to USB Host Controller
Memory Address	0xA0000-0xBFFFF	PCI bus
Memory Address	0xA0000-0xBFFFF	ATI ES1000
Memory Address	0xFA000000-0xFBFFFFFF	PCI standard PCI-to-PCI bridge
Memory Address	0xFA000000-0xFBFFFFFF	PCI standard PCI-to-PCI bridge
Memory Address	0xFA000000-0xFBFFFFFF	HP NC373i Virtual Bus Device
Memory Address	0xF8000000-0xF9FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address	0xF8000000-0xF9FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address	0xF8000000-0xF9FFFFFF	HP NC373i Virtual Bus Device
I/O Port	0x00004000-0x00004FFF	PCI standard PCI-to-PCI bridge
I/O Port	0x00004000-0x00004FFF	Smart Array P400I Controller

[DMA]

Resource	Device	Status
Channel 7	Direct memory access controller	OK

[Forced Hardware]

Device	PNP Device ID	Resource	Device	Status
[I/O]				
0x00000000-0x00000CF7	PCI bus	OK	0x00000C80-0x00000C83	Motherboard resources
0x00000000-0x00000CF7	Direct memory access	OK	0x00000CD4-0x00000CD7	Motherboard resources
0x00000D00-0x0000FFFF	PCI bus	OK	0x00000F50-0x00000F58	Motherboard resources
0x00004000-0x00004FFF	PCI standard PCI-to-PCI bridge	OK	0x00000F00-0x00000F00	Motherboard resources
0x00004000-0x00004FFF	Smart Array P400I Controller	OK	0x00000CA0-0x00000CA1	Motherboard resources
0x00001000-0x0000101F	Standard Universal PCI to USB Host Controller	OK	0x00000CA4-0x00000CA5	Motherboard resources
0x00001020-0x0000103F	Standard Universal PCI to USB Host Controller	OK	0x000002F8-0x000002FF	Motherboard resources
0x00001040-0x0000105F	Standard Universal PCI to USB Host Controller	OK	0x000002F8-0x000002FF	Communications Port
0x00001060-0x0000107F	Standard Universal PCI to USB Host Controller	OK	0x00000CA2-0x00000CA3	HP NULL IPMI Controller
0x00003000-0x000030FF	ATI ES1000	OK	0x00000040-0x00000043	System timer
0x000003B0-0x000003BB	ATI ES1000	OK	0x00000080-0x0000008F	Direct memory access
0x000003C0-0x000003DF	ATI ES1000	OK	0x000000C0-0x000000DF	Direct memory access
0x00002800-0x000028FF	HP ProLiant iLO 2 Legacy Support Function	OK	0x00000061-0x00000061	System speaker
0x00003400-0x000034FF	HP iLO Management Channel Interface Driver	OK	0x00000060-0x00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
0x00003800-0x0000381F	Standard Universal PCI to USB Host Controller	OK	0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
0x00000A79-0x00000A79	ISAPNP Read Data Port	OK	0x0000002E-0x0000002F	Extended IO Bus
0x00000279-0x00000279	ISAPNP Read Data Port	OK	0x0000004E-0x0000004F	Extended IO Bus
0x00000274-0x00000277	ISAPNP Read Data Port	OK	0x00000620-0x0000065F	Extended IO Bus
0x00000070-0x00000077	Motherboard resources	OK	0x00000680-0x0000069F	Extended IO Bus
0x00000408-0x0000040F	Motherboard resources	OK	0x00000600-0x0000061F	Extended IO Bus
0x000004D0-0x000004D1	Motherboard resources	OK	0x00000660-0x0000067F	Extended IO Bus
0x00000020-0x0000003F	Motherboard resources	OK	0x00000300-0x0000030F	Extended IO Bus
0x000000A0-0x000000BF	Motherboard resources	OK	0x000003F8-0x000003FF	Communications Port
0x00000090-0x0000009F	Motherboard resources	OK	0x00000500-0x0000050F	Standard Dual Channel PCI IDE Controller
0x00000050-0x00000053	Motherboard resources	OK	0x000001F0-0x000001F7	Primary IDE Channel
0x00000700-0x0000071F	Motherboard resources	OK	0x000003F6-0x000003F6	Primary IDE Channel
0x00000800-0x0000083F	Motherboard resources	OK	0x00000170-0x00000177	Secondary IDE Channel
0x00000900-0x0000097F	Motherboard resources	OK	0x00000376-0x00000376	Secondary IDE Channel
0x00000010-0x0000001F	Motherboard resources	OK		
			[IRQs]	
			Resource	Device
			IRQ 9	Microsoft ACPI-Compliant System
			IRQ 16	PCI standard PCI-to-PCI bridge
			IRQ 16	Smart Array P400I Controller
			IRQ 16	Standard Universal PCI to USB Host Controller
			IRQ 16	Standard Enhanced PCI to USB Host Controller
			IRQ 17	PCI standard PCI-to-PCI bridge
			IRQ 17	Standard Universal PCI to USB Host Controller
			IRQ 18	PCI standard PCI-to-PCI bridge
			IRQ 18	HP NC373i Virtual Bus Device
			IRQ 18	Standard Universal PCI to USB Host Controller
			IRQ 19	HP NC373i Virtual Bus Device
			IRQ 19	Standard Universal PCI to USB Host Controller
			IRQ 23	ATI ES1000
			IRQ 5	HP ProLiant iLO 2 Legacy Support Function
			IRQ 22	HP iLO Management Channel Interface Driver
			IRQ 22	Standard Universal PCI to USB Host Controller
			IRQ 21	HP ProLiant iLO 2 Management Controller Driver
			IRQ 0	System timer
			IRQ 1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
			IRQ 12	PS/2 Compatible Mouse
			IRQ 4	Communications Port (COM1)
			IRQ 14	Primary IDE Channel
			IRQ 3	Communications Port (COM2)
			[Memory]	
			Resource	Device
			0xA0000-0xBFFFF	PCI bus
			0xA0000-0xBFFFF	ATI ES1000
			0x40000000-0xDFFFFFFF	PCI bus
			0xF0000000-0xFEBFFFFFFF	PCI bus
			0xFDF00000-0xFDFFFFFFFF	PCI standard PCI-to-PCI bridge
			0xFDD00000-0xFDEFFFFFFF	PCI standard PCI-to-PCI bridge
			0xFDE00000-0xFDEFFFFFFF	Smart Array P400I Controller
			0xFDDF0000-0xFDDF0FFF	Smart Array P400I Controller
			0xF8000000-0xF9FFFFFFF	PCI standard PCI-to-PCI bridge
			0xF8000000-0xF9FFFFFFF	PCI standard PCI-to-PCI bridge
			0xF8000000-0xF9FFFFFFF	HP NC373i Virtual Bus Device
			0xFA000000-0xFBFFFFFFF	PCI standard PCI-to-PCI bridge
			0xFA000000-0xFBFFFFFFF	PCI standard PCI-to-PCI bridge

```

OxFA000000-0xFBFFFFFF HP NC373i Virtual Bus
Device OK
OxF7DF0000-0xF7DF03FF Standard Enhanced PCI
to USB Host Controller OK
0xD8000000-0xDFFFFFFF ATI ES1000 OK

0xF7FF0000-0xF7FFFFFF ATI ES1000 OK

0xF7FE0000-0xF7FE01FF HP ProLiant iLO 2
Legacy Support Function OK
0xF7FD0000-0xF7FD07FF HP iLO Management
Channel Interface Driver OK
0xF7FC0000-0xF7FC1FFF HP iLO Management
Channel Interface Driver OK
0xF7F00000-0xF7F7FFFF HP iLO Management
Channel Interface Driver OK
0xF7EF0000-0xF7EF00FF HP ProLiant iLO 2
Management Controller Driver OK
0xE0000000-0xEFFFFFFF Motherboard resources
OK
0xFE000000-0xFEFFFFFF Motherboard resources
OK
0xFED00000-0xFED003FF High precision event
timer OK

```

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
			Creation Date			
c:\windows\system32\sl_anet.acm	Sipro Lab Telecom Inc.	Sipro Lab Telecom Audio Codec	OK	C:\WINDOWS\system32\SLANET.ACM	3.02	84.00 KB (86,016 bytes)
			Creation Date			
c:\windows\system32\msaud32.acm	Microsoft Corporation	Windows Media Audio Codec	OK	C:\WINDOWS\system32\MSAUD32.ACM	8.00.00.4487	288.00 KB (294,912 bytes)
			Creation Date			
c:\windows\system32\msg723.acm	Microsoft Corporation	Windows Media Audio Codec	OK	C:\WINDOWS\system32\MSG723.ACM	5.2.3790.3959	120.00 KB (122,880 bytes)
			Creation Date			
c:\windows\system32\tssoft32.acm	DSP GROUP, INC.	DSP GROUP, INC.	OK	C:\WINDOWS\system32\TSSOFT32.ACM	1.01	9.50 KB (9,728 bytes)
			Creation Date			
c:\windows\system32\msg711.acm	Microsoft Corporation	Windows Media Audio Codec	OK	C:\WINDOWS\system32\MSG711.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	

```

10.00 KB (10,240 bytes) 11/30/2005
6:00 AM
c:\windows\system32\msadp32.acm Microsoft Corporation OK
C:\WINDOWS\system32\MSADP32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
14.50 KB (14,848 bytes) 11/30/2005
6:00 AM
c:\windows\system32\msgsm32.acm Microsoft Corporation OK
C:\WINDOWS\system32\MSGSM32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
20.50 KB (20,992 bytes) 11/30/2005
6:00 AM
c:\windows\system32\imaadp32.acm Microsoft Corporation OK
C:\WINDOWS\system32\IMAADP32.ACM
5.2.3790.0 (srv03_rtm.030324-2048)
15.50 KB (15,872 bytes) 11/30/2005
6:00 AM
c:\windows\system32\l3codeca.acm Fraunhofer Institut Integrierte Schaltungen IIS IIS MPEG Layer-3 Codec OK
C:\WINDOWS\system32\L3CODECA.ACM
9, 0, 0305 284.00 KB (290,816 bytes)
11/30/2005 6:00 AM

```

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size
			Creation Date			
c:\windows\system32\msh261.drv	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSH261.DRV	5.2.3790.3959	184.00 KB (188,416 bytes)
			Creation Date			
c:\windows\system32\tsbyuv.dll	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\TSBYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	8.00 KB (8,192 bytes)
			Creation Date			
c:\windows\system32\msyuv.dll	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	16.50 KB (16,896 bytes)
			Creation Date			
c:\windows\system32\msvidc32.dll	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSVIDC32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	26.50 KB (27,136 bytes)
			Creation Date			
c:\windows\system32\mrle32.dll	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\MSRLE32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	10.50 KB (10,752 bytes)
			Creation Date			
c:\windows\system32\iyuv_32.dll	Microsoft Corporation	Microsoft Corporation	OK	C:\WINDOWS\system32\IYUV_32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	

```

C:\WINDOWS\system32\IYUV_32.DLL
5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
46.50 KB (47,616 bytes) 3/24/2005
1:05 PM
c:\windows\system32\msh263.drv Microsoft Corporation OK
C:\WINDOWS\system32\MSH263.DRV
5.2.3790.3959 288.00 KB (294,912 bytes)
3/24/2005 1:07 PM

```

[CD-ROM]

Item	Value
Drive D:	
Description	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	COMPAQ CD-ROM SN-124
Manufacturer	(Standard CD-ROM drives)
Status	OK
Transfer Rate	Not Available
SCSI Target ID	0
PNP Device ID	IIDE\CDROMCOMPAQ_CD-ROM_SN-124
Driver	c:\windows\system32\drivers\cdrom.sys (5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 51.00 KB (52,224 bytes), 11/30/2005 6:00 AM)

[Sound Device]

Item	Value
Name	ATI ES1000
PNP Device ID	PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_02\4&2014205D&0&18F0
Adapter Type	ATI ES1000 (0x515E), ATI Technologies Inc. compatible
Adapter Description	ATI ES1000
Adapter RAM	32.00 MB (33,554,432 bytes)
Installed Drivers	ati2dvag.dll
Driver Version	6.14.10.6606
INF File	oem11.inf (ati2mtag_RN50 section)
Color Planes	1
Color Table Entries	4294967296
Resolution	1024 x 768 x 60 hertz
Bits/Pixel	32
Memory Address	0xD8000000-0xDFFFFFFF
I/O Port	0x00003000-0x000030FF
Memory Address	0xF7FF0000-0xF7FFFFFF
IRQ Channel	IRQ 23
I/O Port	0x000003B0-0x000003BB
I/O Port	0x000003C0-0x000003DF
Memory Address	0xA0000-0xBFFFF
Driver	c:\windows\system32\drivers\ati2mtag.sys (6.14.10.6606, 1.36 MB (1,431,040 bytes), 8/13/2007 1:53 PM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value

Description USB Human Interface Device
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID
USB\VID_03F0&PID_1027&MI_00\7&2CD6FDA9&0&00
00
Number of Function Keys 12
Driver c:\windows\system32\drivers\hidusb.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 11.50 KB (11,776 bytes), 11/30/2005 6:00 AM)

Description Standard 101/102-Key or Microsoft
Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&2AA4AD3D&0
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver c:\windows\system32\drivers\i8042prt.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 54.50 KB (55,808 bytes), 11/30/2005 6:00 AM)

[Pointing Device]

Item Value

Hardware Type USB Human Interface Device
Number of Buttons 3
Status OK
PNP Device ID
USB\VID_03F0&PID_1027&MI_01\7&2CD6FDA9&0&00
01
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
Driver c:\windows\system32\drivers\hidusb.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 11.50 KB (11,776 bytes), 11/30/2005 6:00 AM)

Hardware Type PS/2 Compatible Mouse
Number of Buttons 3
Status Error
PNP Device ID ACPI\PNP0F13\4&2AA4AD3D&0
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver c:\windows\system32\drivers\i8042prt.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 54.50 KB (55,808 bytes), 11/30/2005 6:00 AM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value

Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 11/3/2009 9:38 AM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 11/3/2009 9:38 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
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 64.00 KB (65,536 bytes), 11/30/2005 6:00 AM)

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 11/3/2009 9:38 AM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\rasppptp.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 58.50 KB (59,904 bytes), 11/30/2005 6:00 AM)

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPOINT\0000
Last Reset 11/3/2009 9:38 AM
Index 4
Service Name Rasppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\rasppoe.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 40.00 KB (40,960 bytes), 11/30/2005 6:00 AM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPOINT\0000
Last Reset 11/3/2009 9:38 AM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 19.50 KB (19,968 bytes), 11/30/2005 6:00 AM)

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 11/3/2009 9:38 AM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Driver c:\windows\system32\drivers\ndiswan.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 87.50 KB
(89,600 bytes), 11/30/2005 6:00 AM)

Name [00000007] HP NC373i Multifunction Gigabit
Server Adapter
Adapter Type Ethernet 802.3
Product Type HP NC373i Multifunction Gigabit
Server Adapter
Installed Yes
PNP Device ID

B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_12\6&30C55FC0&0&20050300
Last Reset 11/3/2009 9:38 AM
Index 7
Service Name l2nd
IP Address 130.172.11.121
IP Subnet 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:21:5A:4C:29:0A

Driver c:\windows\system32\drivers\bxd52x.sys
(4.5.4.0 built by: WinDDK, 54.50 KB (55,808 bytes),
8/10/2007 9:49 AM)

Name [00000008] HP NC373i Multifunction Gigabit
Server Adapter
Adapter Type Ethernet 802.3
Product Type HP NC373i Multifunction Gigabit
Server Adapter
Installed Yes
PNP Device ID

B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_12\6&29511DBC&0&20050500
Last Reset 11/3/2009 9:38 AM
Index 8
Service Name l2nd
IP Address 130.168.40.121, 130.121.208.121
IP Subnet 255.255.0.0, 255.255.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:21:5A:4C:29:0E

Driver c:\windows\system32\drivers\bxd52x.sys
(4.5.4.0 built by: WinDDK, 54.50 KB (55,808 bytes),
8/10/2007 9:49 AM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery Yes	
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes

Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption No	
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery No	
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)

Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption No	
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery No	
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)

Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption Yes	
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP TCP Service Provider
Connectionless Service	No
Guarantees Delivery Yes	
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No

Supports Encryption	Yes
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS
[\Device\NetBT_Tcpip_{337E4A0F-1A8B-4B0D-8AB9-98DB7B9EC7AB}] SEQPACKE	3
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)

Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption No	
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS
[\Device\NetBT_Tcpip_{337E4A0F-1A8B-4B0D-8AB9-98DB7B9EC7AB}] DATAGRAM	3
Connectionless Service	Yes
Guarantees Delivery No	
Guarantees Sequencing	No
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)

Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption No	
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD NetBIOS
[\Device\NetBT_Tcpip_{8DB86463-9958-424F-A4C2-FB3B07FD6B39}] SEQPACKE	0
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	20 bytes
Maximum Message Size	62.50 KB (64,000 bytes)

Message Oriented	Yes
Minimum Address Size	20 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No

Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{8DB86463-9958-424F-A4C2-FB3B07FD6B39}] DATAGRAM 0
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{9F95CA4D-45AE-4E2B-8D26-D0A991E9DDD9}] SEQPACKET 1
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{9F95CA4D-45AE-4E2B-8D26-D0A991E9DDD9}] DATAGRAM 1
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes

Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{D746FA27-DFC0-4D82-B5DF-26123541D6A3}] SEQPACKET 2
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

Name MSAFD NetBIOS
 [\Device\NetBT_Tcpip_{D746FA27-DFC0-4D82-B5DF-26123541D6A3}] DATAGRAM 2
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 20 bytes
 Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
 Minimum Address Size 20 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

[WinSock]

Item	Value
File	c:\windows\system32\winsock.dll
Size	2.80 KB (2,864 bytes)
Version	3.10
File	c:\windows\system32\wsock32.dll
Size	22.00 KB (22,528 bytes)
Version	5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item	Value
Name	Communications Port (COM2)
Status	OK
PNP Device ID	ROOT*PNP0501\1_0_17_1_0_0
Maximum Input Buffer Size	0
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
I/O Port	0x000002F8-0x000002FF
IRQ Channel	IRQ 3
Driver	c:\windows\system32\drivers\serial.sys (5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 64.00 KB (65,536 bytes), 11/30/2005 6:00 AM)

Item	Value
Name	Communications Port (COM1)
Status	OK
PNP Device ID	ACPI\PNP0501\0
Maximum Input Buffer Size	0
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 4
I/O Port 0x000003F8-0x000003FF
Driver c:\windows\system32\drivers\serial.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 64.00 KB
(65,536 bytes), 11/30/2005 6:00 AM)

```

[Parallel]

```
Item Value
```

[Storage]

[Drives]

```

Item Value
Drive C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 33.88 GB (36,381,306,880 bytes)
Free Space 12.54 GB (13,460,373,504 bytes)

```

```

Volume Name
Volume Serial Number 8C06AC55

```

```

Drive D:
Description CD-ROM Disc

```

[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 1
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 2
SCSI Target ID 4
Sectors/Track 32
Size 33.89 GB (36,385,505,280 bytes)
Total Cylinders 8,709
Total Sectors 71,065,440
Total Tracks 2,220,795
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 33.88 GB (36,381,310,976 bytes)

```

```
Partition Starting Offset 16,384 bytes
```

[SCSI]

```

Item Value
Name Smart Array P400I Controller
Manufacturer Hewlett-Packard Company
Status OK
PNP Device ID
PCI\VEN_103C&DEV_3230&SUBSYS_3235103C&REV_0
3\4&EFC3E79&0&0018
Memory Address 0xFDE00000-0xFDEFFFFF
I/O Port 0x00004000-0x00004FFF
Memory Address 0xFDDF0000-0xFDDF0FFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\hpcisss2.sys
(6.6.0.32 Build 5 (x86) built by: buildsrv, 53.30 KB
(54,584 bytes), 12/31/1979 6:00 PM)

```

[IDE]

```

Item Value
Name Standard Dual Channel PCI IDE Controller

```

```

Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID
PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&F9
I/O Port 0x00000500-0x0000050F
Driver c:\windows\system32\drivers\pciide.sys
(5.2.3790.0 (srv03_rtm.030324-2048), 5.50 KB (5,632
bytes), 11/30/2005 6:00 AM)

```

```

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&56E2F28&0&0

```

```

I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver c:\windows\system32\drivers\atapi.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 94.50 KB
(96,768 bytes), 11/30/2005 6:00 AM)

```

```

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI
controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&56E2F28&0&1

```

```

I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
Driver c:\windows\system32\drivers\atapi.sys
(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 94.50 KB
(96,768 bytes), 11/30/2005 6:00 AM)

```

[Printing]

```
Name Driver Port Name Server Name
```

[Problem Devices]

```

Device PNP Device ID Error Code
Standard 101/102-Key or Microsoft Natural PS/2
Keyboard ACPI\PNP0303\4&2AA4AD3D&0 This device
is not present, is not working properly, or does not
have all its drivers installed.
PS/2 Compatible Mouse
ACPI\PNP0F13\4&2AA4AD3D&0 This device
is not present, is not working properly, or does not
have all its drivers installed.

```

[USB]

```

Device PNP Device ID
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_2688&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E8
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_2689&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E9
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_268A&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EA
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_268B&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EB
Standard Enhanced PCI to USB Host Controller
PCI\VEN_8086&DEV_268C&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&EF
Standard Universal PCI to USB Host Controller
PCI\VEN_103C&DEV_3300&SUBSYS_3305103C&REV_0
0\4&2014205D&0&24F0

```

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	Status	Error Control	Accept Pause
atapi	Standard IDE/ESDI Hard Disk Controller	c:\windows\system32\drivers\atapi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
abiosdsk	Abiosdsk Not Available		Kernel Driver	No	Disabled Stopped	OK		
acpi	Microsoft ACPI Driver	c:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No
adpu160m	adpu160m Not Available		Kernel Driver	No	Disabled Stopped	OK		
adpu320	adpu320 Not Available		Kernel Driver	No	Disabled Stopped	OK		
afcmt	afcmt Not Available		Kernel Driver	No	Disabled Stopped	OK		
afd	AFD	c:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	System	Running	OK	Normal No Yes
aic78u2	aic78u2 Not Available		Kernel Driver	No	Disabled Stopped	OK		
aic78xx	aic78xx Not Available		Kernel Driver	No	Disabled Stopped	OK		
aliide	AliIde Not Available		Kernel Driver	No	Disabled Stopped	OK		
alkernel	Altiris Kernel Driver	c:\windows\system32\drivers\alkernel.sys	Kernel Driver	Yes	Manual	Running	OK	Normal No Yes
amdide	AmdIde	c:\windows\system32\drivers\amdide.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No No
arc	arc	c:\windows\system32\drivers\arc.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No No
asynmac	RAS Asynchronous Media Driver	c:\windows\system32\drivers\asynmac.sys	Kernel Driver	No	Manual	Stopped	OK	Normal No No
atdisk	Atdisk Not Available		Kernel Driver	No	Disabled Stopped	OK		
ati2mtag	ati2mtag	c:\windows\system32\drivers\ati2mtag.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore No Yes
atmarpc	ATM ARP Client Protocol	c:\windows\system32\drivers\atmarpc.sys	Kernel Driver	No	Manual	Stopped	OK	Normal No No
audstub	Audio Stub Driver	c:\windows\system32\drivers\audstub.sys	Kernel Driver	Yes	Manual	Running	OK	Normal No Yes
b06bdrv	HP Virtual Bus Device	c:\windows\system32\drivers\bxvbdx.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
beep	Beep	c:\windows\system32\drivers\beep.sys	Kernel Driver	Yes	System	Running	OK	Normal No Yes
cbidf2k	cbidf2k	c:\windows\system32\drivers\cbidf2k.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No No
cd20xrnt	cd20xrnt Not Available		Kernel Driver	No	Disabled Stopped	OK		
cdfs	Cdfs	c:\windows\system32\drivers\cdfs.sys	File System Driver	Yes	Disabled	Running	OK	Normal No Yes
cdrom	CD-ROM Driver	c:\windows\system32\drivers\cdrom.sys	Kernel Driver	Yes	System	Running	OK	Normal No Yes
changer	Changer Not Available		Kernel Driver	No	System Stopped	OK		
clusdisk	Cluster Disk Driver	c:\windows\system32\drivers\clusdisk.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No No
cmdide	CmdIde Not Available		Kernel Driver	No	Disabled Stopped	OK		
cpqarray	Cpqarray Not Available		Kernel Driver	No	Disabled Stopped	OK		
cpqarray2	cpqarray2 Not Available		Kernel Driver	No	Disabled Stopped	OK		
cpqcidrv	HP iLO Management Channel Interface Driver	c:\windows\system32\drivers\cpqcidrv.sys	Kernel Driver	Yes	Manual	Running	OK	Normal No Yes
cpqcissm	cpqcissm Not Available		Kernel Driver	No	Disabled Stopped	OK		
cpqfcalm	cpqfcalm Not Available		Kernel Driver	No	Disabled Stopped	OK		
crtdisk	CRC Disk Filter Driver	c:\windows\system32\drivers\crtdisk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
dac960nt	dac960nt Not Available		Kernel Driver	No	Disabled Stopped	OK		
dellcerc	dellcerc Not Available		Kernel Driver	No	Disabled Stopped	OK		
dfsdriver	DfsDriver	c:\windows\system32\drivers\dfs.sys	File System Driver	Yes	Boot	Running	OK	Normal No Yes
disk	Disk Driver	c:\windows\system32\drivers\disk.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
dmboot	dmboot	c:\windows\system32\drivers\dmboot.sys	Kernel Driver	No	Disabled	Stopped	OK	Normal No No
dmio	Logical Disk Manager Driver	c:\windows\system32\drivers\dmio.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
dmload	dmload	c:\windows\system32\drivers\dmload.sys	Kernel Driver	Yes	Boot	Running	OK	Normal No Yes
dpti2o	dpti2o Not Available		Kernel Driver	No	Disabled Stopped	OK		
elxstor	elxstor Not Available		Kernel Driver	No	Disabled Stopped	OK		
fastfat	Fastfat	c:\windows\system32\drivers\fastfat.sys	File System Driver	No	Disabled			

	Stopped	OK	Normal	No	No		Running	OK	Normal	No	Yes		isapnp	PnP ISA/EISA Bus Driver c:\windows\system32\drivers\isapnp.sys Kernel Driver Yes Boot Running OK Critical No Yes
fdc	Fdc c:\windows\system32\drivers\fdc.sys Kernel Driver No System Stopped OK Ignore No No					i2omgmt	i2omgmt Not Available No System Stopped OK Normal No No						i2omp	i2omp Not Available No Disabled Stopped OK Normal No No
fips	Fips c:\windows\system32\drivers\fips.sys Kernel Driver Yes System Running OK Normal No Yes					i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver c:\windows\system32\drivers\i8042prt.sys Kernel Driver Yes System Running OK Normal No Yes						kbdclass	Keyboard Class Driver c:\windows\system32\drivers\kbdclass.sys Kernel Driver Yes System Running OK Normal No Yes
flpydisk	Flpydisk c:\windows\system32\drivers\flpydisk.sys Kernel Driver No System Stopped OK Ignore No No					iirsp	iirsp Not Available No Disabled Stopped OK Normal No No						kbdhid	Keyboard HID Driver c:\windows\system32\drivers\kbdhid.sys Kernel Driver Yes System Running OK Ignore No Yes
fltmgr	FltMgr c:\windows\system32\drivers\fltmgr.sys File System Driver Yes Boot Running OK Normal No Yes					imapi	CD-Burning Filter Driver c:\windows\system32\drivers\imapi.sys Kernel Driver No System Stopped OK Normal No No						ksecdd	KSecDD c:\windows\system32\drivers\ksecdd.sys Kernel Driver Yes Boot Running OK Normal No Yes
ftdisk	Volume Manager Driver c:\windows\system32\drivers\ftdisk.sys Kernel Driver Yes Boot Running OK Normal No Yes					intelide	IntelIde Not Available No Disabled Stopped OK Normal No No						l2nd	HP NC370 Multifunction Gigabit Server Adapter c:\windows\system32\drivers\bxnd52x.sys Kernel Driver Yes Manual Running OK Normal No Yes
gpc	Generic Packet Classifier c:\windows\system32\drivers\msgpc.sys Kernel Driver Yes Manual Running OK Normal No Yes					intelppm	Intel Processor Driver c:\windows\system32\drivers\intelppm.sys Kernel Driver Yes Manual Running OK Normal No Yes						lp6nds35	lp6nds35 Not Available No Disabled Stopped OK Normal No No
hidusb	Microsoft HID Class Driver c:\windows\system32\drivers\hidusb.sys Kernel Driver Yes Manual Running OK Ignore No Yes					ip6fw	IPv6 Windows Firewall Driver c:\windows\system32\drivers\ip6fw.sys Kernel Driver No Manual Stopped OK Normal No No						mnmdd	mnmdd c:\windows\system32\drivers\mnmdd.sys Kernel Driver Yes System Running OK Ignore No Yes
hpcisss	hpcisss c:\windows\system32\drivers\hpcisss.sys Kernel Driver No Disabled Stopped OK Normal No No					ipfilterdriver	IP Traffic Filter Driver c:\windows\system32\drivers\ipfltdrv.sys Kernel Driver No Manual Stopped OK Normal No No						modem	Modem c:\windows\system32\drivers\modem.sys Kernel Driver No Manual Stopped OK Ignore No No
hpcisss2	HpCISSs2 c:\windows\system32\drivers\hpcisss2.sys Kernel Driver Yes Boot Running OK Normal No Yes					ipinip	IP in IP Tunnel Driver c:\windows\system32\drivers\ipinip.sys Kernel Driver No Manual Stopped OK Normal No No						mouclass	Mouse Class Driver c:\windows\system32\drivers\mouclass.sys Kernel Driver Yes System Running OK Normal No Yes
hpn	hpn Not Available No Disabled Stopped OK Normal No No					ipnat	IP Network Address Translator c:\windows\system32\drivers\ipnat.sys Kernel Driver No Manual Stopped OK Normal No No						mouhid	Mouse HID Driver c:\windows\system32\drivers\mouhid.sys Kernel Driver Yes Manual Running OK Ignore No Yes
hpqilo2	hpqilo2 c:\windows\system32\drivers\hpqilo2.sys Kernel Driver Yes Manual Running OK Normal No Yes					ipsec	IPSEC driver c:\windows\system32\drivers\ipsec.sys Kernel Driver Yes System Running OK Normal No Yes						mountmgr	Mount Point Manager c:\windows\system32\drivers\mountmgr.sys Kernel Driver Yes Boot Running OK Normal No Yes
hpt3xx	hpt3xx Not Available No Disabled Stopped OK Normal No No					ipsraidn	ipsraidn Not Available No Disabled Stopped OK Normal No No						mraid35x	mraid35x Not Available No Disabled Stopped OK Normal No No
http	HTTP c:\windows\system32\drivers\http.sys Kernel Driver Yes Manual					irenum	IR Enumerator Service c:\windows\system32\drivers\irenum.sys Kernel Driver No Manual Stopped OK Normal No No						mrxdav	WebDav Client Redirector c:\windows\system32\drivers\mrxdav.sys File System Driver No Manual Stopped OK Normal No No

mrxsmbr	MRXSMB c:\windows\system32\drivers\mrxsmbr.sys File System Driver Yes System Running OK Normal No Yes	Running OK Normal No Yes	Kernel Driver Yes Manual Running OK Normal No Yes
msfs	Msfss c:\windows\system32\drivers\msfss.sys File System Driver Yes System Running OK Normal No Yes	ntfs Ntfs c:\windows\system32\drivers\ntfs.sys File System Driver Yes Disabled Running OK Normal No Yes	ql1080 ql1080 Not Available Kernel Driver No Disabled Stopped OK Normal No No
mssmbios	Microsoft System Management BIOS Driver c:\windows\system32\drivers\mssmbios.sys Kernel Driver Yes Manual Running OK Normal No Yes	null Null c:\windows\system32\drivers\null.sys Kernel Driver Yes System Running OK Normal No Yes	ql10wmt ql10wmt Not Available Kernel Driver No Disabled Stopped OK Normal No No
mup	Mup c:\windows\system32\drivers\mup.sys File System Driver Yes Boot Running OK Normal No Yes	parport Parport c:\windows\system32\drivers\parport.sys Kernel Driver No Manual Stopped OK Ignore No No	ql12160 ql12160 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ndis	NDIS System Driver c:\windows\system32\drivers\ndis.sys Kernel Driver Yes Boot Running OK Normal No Yes	partmgr Partition Manager c:\windows\system32\drivers\partmgr.sys Kernel Driver Yes Boot Running OK Normal No Yes	ql1240 ql1240 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ndistapi	Remote Access NDIS TAPI Driver c:\windows\system32\drivers\ndistapi.sys Kernel Driver Yes Manual Running OK Normal No Yes	pci PCI Bus Driver c:\windows\system32\drivers\pci.sys Kernel Driver Yes Boot Running OK Critical No Yes	ql1280 ql1280 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ndisuiop	NDIS Usermode I/O Protocol c:\windows\system32\drivers\ndisuiop.sys Kernel Driver No Manual Stopped OK Normal No No	pciide PCIIDE c:\windows\system32\drivers\pciide.sys Kernel Driver Yes Boot Running OK Normal No Yes	ql2100 ql2100 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ndiswan	Remote Access NDIS WAN Driver c:\windows\system32\drivers\ndiswan.sys Kernel Driver Yes Manual Running OK Normal No Yes	pcmcia Pcmcia c:\windows\system32\drivers\pcmcia.sys Kernel Driver No Disabled Stopped OK Normal No No	ql2200 ql2200 Not Available Kernel Driver No Disabled Stopped OK Normal No No
ndproxy	NDIS Proxy c:\windows\system32\drivers\ndproxy.sys Kernel Driver Yes Manual Running OK Normal No Yes	pdcomp PDCOMP Not Available Kernel Driver No Manual Stopped OK Ignore No No	ql2300 ql2300 Not Available Kernel Driver No Disabled Stopped OK Normal No No
netbios	NetBIOS Interface c:\windows\system32\drivers\netbios.sys File System Driver Yes System Running OK Normal No Yes	pdreli PDRELI Not Available Kernel Driver No Manual Stopped OK Ignore No No	rasacd Remote Access Auto Connection Driver c:\windows\system32\drivers\rasacd.sys Kernel Driver Yes System Running OK Normal No Yes
netbt	NetBios over Tcpip c:\windows\system32\drivers\netbt.sys Kernel Driver Yes System Running OK Normal No Yes	pdrframe PDRFRAME Not Available Kernel Driver No Manual Stopped OK Ignore No No	rasl2tp WAN Miniport (L2TP) c:\windows\system32\drivers\rasl2tp.sys Kernel Driver Yes Manual Running OK Normal No Yes
nfrd960	nfrd960 Not Available Kernel Driver No Disabled Stopped OK Normal No No	perc2 perc2 Not Available Kernel Driver No Disabled Stopped OK Normal No No	raspppoe Remote Access PPPOE Driver c:\windows\system32\drivers\raspppoe.sys Kernel Driver Yes Manual Running OK Normal No Yes
npfs	Npfs c:\windows\system32\drivers\npfs.sys File System Driver Yes System	perc2hib perc2hib Not Available Kernel Driver No Disabled Stopped OK Normal No No	raspti Direct Parallel c:\windows\system32\drivers\raspti.sys Kernel Driver Yes Manual Running OK Normal No Yes
		pptpminiport WAN Miniport (PPTP) c:\windows\system32\drivers\raspptp.sys Kernel Driver Yes Manual Running OK Normal No Yes	rdbss Rdbss c:\windows\system32\drivers\rdbss.sys File System Driver Yes System Running OK Normal No Yes
		ptilink Direct Parallel Link Driver c:\windows\system32\drivers\ptilink.sys	rdpcdd RDPcDD c:\windows\system32\drivers\rdpcdd.sys Kernel Driver Yes System Running OK Ignore No Yes
			rdpdr Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys Kernel Driver Yes Manual Running OK Normal No Yes

rdpwd	RDPWD c:\windows\system32\drivers\rdpwd.sys Kernel Driver Yes Manual Running OK Ignore No Yes
redbook	Digital CD Audio Playback Filter Driver c:\windows\system32\drivers\redbook.sys Kernel Driver Yes System Running OK Normal No Yes
secdrv	Secdrv c:\windows\system32\drivers\secdrv.sys Kernel Driver No Manual Stopped OK Normal No No
serenum	Serenum Filter Driver c:\windows\system32\drivers\serenum.sys Kernel Driver 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
sfloppy	Sfloppy c:\windows\system32\drivers\sfloppy.sys Kernel Driver No System Stopped OK Ignore No No
simbad	Simbad Not Available Kernel Driver No Disabled Stopped OK Normal No No
srv	Srv c:\windows\system32\drivers\srv.sys File System Driver Yes Manual Running OK Normal No Yes
startdss Driver	HP ProLiant Virtual Install Disk Support c:\windows\system32\drivers\startdss.sys Kernel Driver No Disabled Stopped OK Normal No No
swenum	Software Bus Driver c:\windows\system32\drivers\swenum.sys Kernel Driver Yes Manual Running OK Normal No Yes
symc810	symc810 Not Available Kernel Driver No Disabled Stopped OK Normal No No
symc8xx	symc8xx Not Available Kernel Driver No Disabled Stopped OK Normal No No
symmpi	symmpi Not Available Kernel Driver No Disabled Stopped OK Normal No No
sym_hi	sym_hi Not Available Kernel Driver No Disabled Stopped OK Normal No No

sym_u3	sym_u3 Not Available Kernel Driver No Disabled Stopped OK Normal No No
tcpip	TCP/IP Protocol Driver c:\windows\system32\drivers\tcpip.sys Kernel Driver Yes System Running OK Normal No Yes
tdpipe	TDPIPE c:\windows\system32\drivers\tdpipe.sys Kernel Driver No Manual Stopped OK Ignore No No
tdtcp	TDTCP c:\windows\system32\drivers\tdtcp.sys Kernel Driver Yes Manual Running OK Ignore No Yes
termdd	Terminal Device Driver c:\windows\system32\drivers\termdd.sys Kernel Driver Yes System Running OK Normal No Yes
toside	TosIde Not Available Kernel Driver No Disabled Stopped OK Normal No No
udfs	Udfs c:\windows\system32\drivers\udfs.sys File System Driver No Disabled Stopped OK Normal No No
ultra	ultra Not Available Kernel Driver No Disabled Stopped OK Normal No No
update	Microcode Update Driver c:\windows\system32\drivers\update.sys Kernel Driver Yes Manual Running OK Normal No Yes
usbccgp	Microsoft USB Generic Parent Driver c:\windows\system32\drivers\usbccgp.sys Kernel Driver Yes Manual Running OK Normal No Yes
usbehci Miniport	Microsoft USB 2.0 Enhanced Host Controller Driver c:\windows\system32\drivers\usbehci.sys Kernel Driver Yes Manual Running OK Normal No Yes
usbhub	Microsoft USB Standard Hub Driver c:\windows\system32\drivers\usbhub.sys Kernel Driver Yes Manual Running OK Normal No Yes
usbstor	USB Mass Storage Driver c:\windows\system32\drivers\usbstor.sys Kernel Driver No Manual Stopped OK Normal No No
usbuhci Miniport	Microsoft USB Universal Host Controller Driver

vga	vga c:\windows\system32\drivers\vgapnp.sys Kernel Driver No Manual Stopped OK Ignore No No
vgasave	VGA Display Controller. c:\windows\system32\drivers\vga.sys Kernel Driver Yes System Running OK Ignore No Yes
viaide	ViaIde Not Available Kernel Driver No Disabled Stopped OK Normal No No
volsnap	Storage volumes c:\windows\system32\drivers\volsnap.sys Kernel Driver Yes Boot Running OK Normal No Yes
wanarp	Remote Access IP ARP Driver c:\windows\system32\drivers\wanarp.sys Kernel Driver Yes Manual Running OK Normal No Yes
wdf01000	Wdf01000 c:\windows\system32\drivers\wdf01000.sys Kernel Driver Yes Boot Running OK Normal No Yes
wdica	WDICA Not Available Kernel Driver No Manual Stopped OK Ignore No No
wlbs	Network Load Balancing c:\windows\system32\drivers\wlbs.sys Kernel Driver No Manual Stopped OK Normal No No
[Signed Drivers]	
Device Name	Signed Device Class
Driver Version	Driver Date
Manufacturer	INF Name Driver Name
Device ID	
Communications Port	Yes PORTS 5.2.3790.0
10/1/2002 (Standard port types)	
msports.inf	Not Available
ROOT*PNP0501\1_0_17_1_0_0	
Microsoft System Management BIOS Driver	Yes
SYSTEM 5.2.3790.1830	10/1/2002
(Standard system devices)	machine.inf
Not Available	ROOT\SYSTEM\0002
Microcode Update Device	Yes SYSTEM
5.2.3790.1830	10/1/2002 (Standard
system devices)	machine.inf Not Available
ROOT\SYSTEM\0001	
Plug and Play Software Device Enumerator	Yes
SYSTEM 5.2.3790.1830	10/1/2002

system devices) machine.inf Not Available
 ROOT\DMIO\0000
 ACPI Fixed Feature Button Yes SYSTEM
 5.2.3790.1830 10/1/2002 (Standard
 system devices) machine.inf Not Available
 ACPI\FIXEDBUTTON\2&DABA3FF&0
 ACPI Thermal Zone Yes SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\THERMALZONE\THM0
 Secondary IDE Channel Yes HDC
 5.2.3790.1830 10/1/2002 (Standard IDE
 ATA/ATAPI controllers) mshdc.inf Not Available
 PCIIDE\IDECHANNEL\4&56E2F28&0&1
 CD-ROM Drive Yes CDROM 5.2.3790.0
 10/1/2002 (Standard CD-ROM drives)
 cdrom.inf Not Available
 IDE\CDROMCOMPAQ_CD-ROM_SN-
 124_____N104_____5&5FD9AC6&0&0.0.0
 Primary IDE Channel Yes HDC 5.2.3790.1830
 10/1/2002 (Standard IDE ATA/ATAPI
 controllers) mshdc.inf Not Available
 PCIIDE\IDECHANNEL\4&56E2F28&0&0
 Standard Dual Channel PCI IDE Controller Yes
 HDC 5.2.3790.1830 10/1/2002
 (Standard IDE ATA/ATAPI controllers)
 mshdc.inf Not Available
 PCI\VEN_8086&DEV_269E&SUBSYS_31FE103C&REV_0
 9\3&61AAA01&0&F9
 Communications Port Yes PORTS 5.2.3790.0
 10/1/2002 (Standard port types)
 msports.inf Not Available
 ACPI\PNP0501\0
 Extended IO Bus Yes SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0A06\4&2AA4AD3D&0
 PS/2 Compatible Mouse Yes MOUSE
 5.2.3790.1830 10/1/2002 Microsoft
 msmouse.inf Not Available
 ACPI\PNP0F13\4&2AA4AD3D&0
 Standard 101/102-Key or Microsoft Natural PS/2
 Keyboard Yes KEYBOARD 5.2.3790.0
 10/1/2002 (Standard keyboards)
 keyboard.inf Not Available
 ACPI\PNP0303\4&2AA4AD3D&0
 System speaker Yes SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0800\4&2AA4AD3D&0
 Direct memory access controller Yes
 SYSTEM 5.2.3790.1830 10/1/2002
 (Standard system devices) machine.inf
 Not Available
 ACPI\PNP0200\4&2AA4AD3D&0
 High precision event timer Yes SYSTEM
 5.2.3790.3959 10/1/2002 (Standard
 system devices) machine.inf Not Available
 ACPI\PNP0103\0
 System timer Yes SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)

machine.inf Not Available
 ACPI\PNP0100\4&2AA4AD3D&0
 HP NULL IPMI Controller Yes SYSTEM
 1.0.0.0 1/1/2004 Hewlett-Packard Company
 oeml2.inf Not Available
 ACPI\IPI0001\0
 Motherboard resources Yes SYSTEM
 5.2.3790.1830 10/1/2002 (Standard
 system devices) machine.inf Not Available
 ACPI\PNP0C02\0
 ISAPNP Read Data Port Yes SYSTEM
 5.2.3790.1830 10/1/2002 (Standard
 system devices) machine.inf Not Available
 ISAPNP\READDATAPORT\0
 PCI standard ISA bridge Yes SYSTEM
 5.2.3790.1830 10/1/2002 (Standard
 system devices) machine.inf Not Available
 PCI\VEN_8086&DEV_2670&SUBSYS_00000000&REV_0
 9\3&61AAA01&0&F8
 HP ProLiant iLO 2 Management Controller Driver Yes
 SYSTEM 1.3.0.0 3/30/2007 Hewlett-
 Packard Company oem9.inf Not Available
 PCI\VEN_103C&DEV_3302&SUBSYS_3305103C&REV_0
 0\4&2014205D&0&26F0
 Generic USB Hub Yes USB 5.2.3790.1830
 10/1/2002 (Generic USB Hub) usb.inf Not
 Available USB\VID_03F0&PID_1327\6&18FFBC52&0&2
 HID-compliant mouse Yes MOUSE 5.2.3790.1830
 10/1/2002 Microsoft msmouse.inf Not
 Available
 HID\VID_03F0&PID_1027&MI_01\8&25B103E6&0&00
 00
 USB Human Interface Device Yes HIDCLASS
 5.2.3790.0 10/1/2002 (Standard
 system devices) input.inf Not Available
 USB\VID_03F0&PID_1027&MI_01\7&2CD6FDA9&0&00
 01
 HID Keyboard Device Yes KEYBOARD 5.2.3790.0
 10/1/2002 (Standard keyboards)
 keyboard.inf Not Available
 HID\VID_03F0&PID_1027&MI_00\8&DED77A1&0&000
 0
 USB Human Interface Device Yes HIDCLASS
 5.2.3790.0 10/1/2002 (Standard
 system devices) input.inf Not Available
 USB\VID_03F0&PID_1027&MI_00\7&2CD6FDA9&0&00
 00
 USB Composite Device Yes USB
 5.2.3790.1830 10/1/2002 (Standard USB
 Host Controller) usb.inf Not Available
 USB\VID_03F0&PID_1027\6&18FFBC52&0&1
 USB Root Hub Yes USB 5.2.3790.1830
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\5&26BC3420&0
 Standard Universal PCI to USB Host Controller Yes
 USB 5.2.3790.1830 10/1/2002
 (Standard USB Host Controller)
 usbport.inf Not Available
 PCI\VEN_8086&DEV_268A&SUBSYS_31FE103C&REV_0
 0\4&2014205D&0&24F0

HP iLO Management Channel Interface Driver Yes
 MULTIFUNCTION 1.12.0.0 6/22/2007
 Hewlett-Packard Company oem4.inf Not
 Available
 PCI\VEN_0E11&DEV_B204&SUBSYS_3305103C&REV_0
 3\4&2014205D&0&22F0
 HP ProLiant iLO 2 Legacy Support Function Yes
 SYSTEM 1.3.0.0 3/30/2007 Hewlett-
 Packard Company oem9.inf Not Available
 PCI\VEN_0E11&DEV_B203&SUBSYS_3305103C&REV_0
 3\4&2014205D&0&20F0
 Default Monitor Yes MONITOR 5.1.2001.0
 6/6/2001 (Standard monitor types)
 monitor.inf Not Available
 DISPLAY\DEFAULT_MONITOR\5&E64F3B&0&10000000
 &01&03
 Default Monitor Yes MONITOR 5.1.2001.0
 6/6/2001 (Standard monitor types)
 monitor.inf Not Available
 DISPLAY\DEFAULT_MONITOR\5&E64F3B&0&10000001
 &01&03
 ATI ES1000 Yes DISPLAY 8.24.3.0
 4/5/2006 ATI Technologies Inc.
 oem11.inf Not Available
 PCI\VEN_1002&DEV_515E&SUBSYS_31FB103C&REV_0
 2\4&2014205D&0&18F0
 Intel(R) 82801 PCI Bridge - 244E Yes
 SYSTEM 5.2.3790.3959 10/1/2002
 Intel machine.inf Not Available
 PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_D
 9\3&61AAA01&0&F0
 USB Root Hub Yes USB 5.2.3790.1830
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB20\4&392538C3&0
 Standard Enhanced PCI to USB Host Controller Yes
 USB 5.2.3790.1830 10/1/2002
 (Standard USB Host Controller)
 usbport.inf Not Available
 PCI\VEN_8086&DEV_268C&SUBSYS_31FE103C&REV_0
 9\3&61AAA01&0&EF
 USB Root Hub Yes USB 5.2.3790.1830
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&41C0314&0
 Standard Universal PCI to USB Host Controller Yes
 USB 5.2.3790.1830 10/1/2002
 (Standard USB Host Controller)
 usbport.inf Not Available
 PCI\VEN_8086&DEV_268B&SUBSYS_31FE103C&REV_0
 9\3&61AAA01&0&EB
 USB Root Hub Yes USB 5.2.3790.1830
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&A54F890&0
 Standard Universal PCI to USB Host Controller Yes
 USB 5.2.3790.1830 10/1/2002
 (Standard USB Host Controller)
 usbport.inf Not Available
 PCI\VEN_8086&DEV_268A&SUBSYS_31FE103C&REV_0
 9\3&61AAA01&0&EA

```

USB Root Hub Yes USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf Not Available
USB\ROOT_HUB\4&37897620&0
Standard Universal PCI to USB Host Controller Yes
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_2689&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E9
USB Root Hub Yes USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf Not Available
USB\ROOT_HUB\4&7353027&0
Standard Universal PCI to USB Host Controller Yes
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller)
usbport.inf Not Available
PCI\VEN_8086&DEV_2688&SUBSYS_31FE103C&REV_0
9\3&61AAA01&0&E8
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F6&SUBSYS_00000000&REV_B
1\3&61AAA01&0&B0
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F5&SUBSYS_00000000&REV_B
1\3&61AAA01&0&A8
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F3&SUBSYS_00000000&REV_B
1\3&61AAA01&0&98
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F1&SUBSYS_00000000&REV_B
1\3&61AAA01&0&88
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_B
1\3&61AAA01&0&82
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_B
1\3&61AAA01&0&81
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_B
1\3&61AAA01&0&80
HP NC373i Multifunction Gigabit Server Adapter Yes
NET 4.5.4.0 8/12/2008 Hewlett-
Packard Company oem16.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_12\6&29511DBC&0&20050500
HP NC373i Virtual Bus Device Yes SYSTEM
4.4.24.0 8/12/2008 Hewlett-Packard Company

```

```

oem19.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_7038103C&REV_1
2\5&3687280D&0&000038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C
3\4&8C20058&0&0038
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E7&SUBSYS_00000000&REV_B
1\3&61AAA01&0&38
HP NC373i Multifunction Gigabit Server Adapter Yes
NET 4.5.4.0 8/12/2008 Hewlett-
Packard Company oem16.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_7038103C&R
EV_12\6&30C55FC0&0&20050300
HP NC373i Virtual Bus Device Yes SYSTEM
4.4.24.0 8/12/2008 Hewlett-Packard Company
oem19.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_7038103C&REV_1
2\5&20B00FF&0&000030
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C
3\4&79C23&0&0030
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E6&SUBSYS_00000000&REV_B
1\3&61AAA01&0&30
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E5&SUBSYS_00000000&REV_B
1\3&61AAA01&0&28
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25F8&SUBSYS_00000000&REV_B
1\3&61AAA01&0&20
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_HP&PROD_LOGICAL_VOLUME&REV_4.
12\5&526D07C&0&000400
HP Virtual LUN Yes SYSTEM 5.2.3790.3959
10/1/2002 Compaq scsidev.inf Not
Available
SCSI\OTHER&VEN_COMPAQ&PROD_SCSI_COMMUNICATE
&REV_CIS2\5&526D07C&0&000000
Smart Array P400I Controller Yes SCSIADAPTER
6.6.0.32 3/20/2007 Hewlett-Packard Company
oem10.inf Not Available

```

```

PCI\VEN_103C&DEV_3230&SUBSYS_3235103C&REV_0
3\4&EFC3E79&0&0018
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E3&SUBSYS_00000000&REV_B
1\3&61AAA01&0&18
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_350C&SUBSYS_00000000&REV_0
1\4&641DA44&0&0310
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3518&SUBSYS_00000000&REV_0
1\5&38BD847A&0&100010
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3514&SUBSYS_00000000&REV_0
1\5&38BD847A&0&080010
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3510&SUBSYS_00000000&REV_0
1\5&38BD847A&0&000010
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_3500&SUBSYS_00000000&REV_0
1\4&641DA44&0&0010
PCI standard PCI-to-PCI bridge Yes
SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf
Not Available
PCI\VEN_8086&DEV_25E2&SUBSYS_00000000&REV_B
1\3&61AAA01&0&10
PCI standard host CPU bridge Yes SYSTEM
5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25D8&SUBSYS_00000000&REV_B
1\3&61AAA01&0&00
PCI bus Yes SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A03\2&DABA3FF&0
Intel Processor Yes PROCESSOR 5.2.3790.3959
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_6_MODEL_23\_3
Intel Processor Yes PROCESSOR 5.2.3790.3959
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL_-
_X86_FAMILY_6_MODEL_23\_2

```

```

Intel Processor Yes PROCESSOR 5.2.3790.3959
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL-_
_X86_FAMILY_6_MODEL_23\_1
Intel Processor Yes PROCESSOR 5.2.3790.3959
10/1/2002 Intel cpu.inf Not Available
ACPI\GENUINEINTEL-_
_X86_FAMILY_6_MODEL_23\_0
Microsoft ACPI-Compliant System Yes
SYSTEM 5.2.3790.0 10/1/2002
Microsoft acpi.inf Not Available
ACPI_HAL\PNP0C08\0
ACPI Multiprocessor PC Yes COMPUTER
5.2.3790.1830 10/1/2002 (Standard
computers) hal.inf Not Available
ROOT\ACPI_HAL\0000
Not Available Not Available Not Available
Not Available Not Available Not Available
Available Not Available Not Available
HTREE\ROOT\0

```

[Environment Variables]

```

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%SystemR
oot%\System32\Wbem;C:\Program Files\Microsoft SQL
Server\80\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\90\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\90\DTS\Binn\;C:\Program Files\Microsoft SQL
Server\90\Tools\Binn\WShell\Common7\IDE\;C:\Program
Files\Microsoft Visual Studio
9\Common7\IDE\PrivateAssemblies\ <SYSTEM>
windir %SystemRoot% <SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 23
Stepping 6, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 1706 <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
ClusterLog C:\WINDOWS\Cluster\cluster.log
<SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
lib C:\Program Files\SQLXML 4.0\bin\
<SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE

```

```

TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp
CL121\Administrator
TMP %USERPROFILE%\Local Settings\Temp
CL121\Administrator

```

[Print Jobs]

Document	Size	Owner	Notify	Status
	Time Submitted		Start Time	
	Until Time		Elapsed Time	
	Pages Printed		Job ID	Priority
	Parameters		Driver	Print
Processor	Host	Print Queue	Data Type	Name

[Network Connections]

Local Name	Remote Name	Type
Status	User Name	

[Running Tasks]

Name	Path	Process ID	Priority	Min
Working Set	Max Working Set	Start Time		
	Version	Size	File Date	
system idle process	Not Available	Not Available	0	0
Available	Not Available	Not Available	Not Available	Not Available
system	Not Available	4	8	0
1380	Not Available	Not Available	Not Available	
smss.exe	Not Available	Not Available	Not Available	
1380	11/3/2009 9:38 AM	Not Available	Not Available	
csrss.exe	Not Available	Not Available	Not Available	
1380	11/3/2009 9:38 AM	Not Available	Not Available	
winlogon.exe	c:\windows\system32\winlogon.exe	412	13	200
11/3/2009 9:38 AM	5.2.3790.3959	516.00 KB (528,384 bytes)	4/15/2008 12:37 PM	
services.exe	c:\windows\system32\services.exe	460	9	200
11/3/2009 9:38 AM	5.2.3790.3959	108.50 KB (111,104 bytes)	11/30/2005 6:00 AM	
lsass.exe	c:\windows\system32\lsass.exe	472	9	200
11/3/2009 9:38 AM	5.2.3790.0 (srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	11/30/2005	
6:00 AM				
svchost.exe	c:\windows\system32\svchost.exe	648	8	200
11/3/2009 9:38 AM	5.2.3790.3959	14.50 KB (14,848 bytes)	4/15/2008 12:38 PM	
svchost.exe	Not Available	716	8	
svchost.exe	Not Available	Not Available		

11/3/2009 9:38 AM	Not Available	Not Available	Not Available	Not Available
svchost.exe	Not Available	776	8	
11/3/2009 9:38 AM	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	800	8	200
11/3/2009 9:38 AM	5.2.3790.3959	14.50 KB (14,848 bytes)	4/15/2008 12:38 PM	
msdtc.exe	Not Available	904	8	Not Available
11/3/2009 9:38 AM	Not Available	11/3/2009 9:38 AM	Not Available	Not Available
aclient.exe	c:\program files\altiris\aclient\aclient.exe	1008	8	
200	1380	11/3/2009 9:38 AM		
6.9.164	5.10 MB (5,349,452 bytes)	8/13/2007 3:16 PM		
svchost.exe	c:\windows\system32\svchost.exe	1092	8	200
11/3/2009 9:38 AM	5.2.3790.3959	14.50 KB (14,848 bytes)	4/15/2008 12:38 PM	
inetinfo.exe	c:\windows\system32\inetrv\inetinfo.exe	1176	8	200
11/3/2009 9:38 AM	6.0.3790.3959	14.00 KB (14,336 bytes)	4/15/2008 12:39 PM	
svchost.exe	Not Available	1232	8	
Not Available	Not Available	Not Available	Not Available	Not Available
rsys.exe	Not Available	1256	8	Not Available
11/3/2009 9:38 AM	Not Available	11/3/2009 9:38 AM	Not Available	Not Available
sysdown.exe	c:\windows\system32\sysdown.exe	1284	8	200
11/3/2009 9:38 AM	1.1.0.0 built by:	6.50 KB (6,656 bytes)	8/13/2007 1:52 PM	
svchost.exe	Not Available	1348	8	
Not Available	Not Available	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	1524	8	200
11/3/2009 9:38 AM	5.2.3790.3959	14.50 KB (14,848 bytes)	4/15/2008 12:38 PM	
svchost.exe	c:\windows\system32\svchost.exe	1668	8	200
11/3/2009 9:38 AM	5.2.3790.3959	14.50 KB (14,848 bytes)	4/15/2008 12:38 PM	
wmiiprvse.exe	Not Available	1908	8	
Not Available	Not Available	Not Available	Not Available	Not Available
11/3/2009 9:40 AM	Not Available	Not Available	Not Available	Not Available
Available	Not Available			

```

logon.scr Not Available      144      4      Not
Available Not Available    11/3/2009 9:48 AM Not
Available Not Available    Not Available
msinfo32.exe c:\program files\common
files\microsoft shared\msinfo\msinfo32.exe
1776      8      200      1380
11/3/2009 12:29 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 42.00 KB (43,008 bytes)
8/7/2007 2:52 PM
wmiprvse.exe Not Available      1320      8
Not Available Not Available
11/3/2009 12:29 PM Not Available Not
Available Not Available

```

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
winlogon	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	516.00 KB (528,384 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\winlogon.exe
ntdll	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	747.50 KB (765,440 bytes)	11/30/2005	Microsoft Corporation	c:\windows\system32\ntdll.dll
kernel32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1,013.00 KB (1,037,312 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\kernel32.dll
advapi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	604.00 KB (618,496 bytes)	11/30/2005	Microsoft Corporation	c:\windows\system32\advapi32.dll
rpcrt4	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	627.00 KB (642,048 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\rpcrt4.dll
secur32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	63.50 KB (65,024 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\secur32.dll
crypt32	5.131.3790.3959 (srv03_sp2_rtm.070216-1710)	581.50 KB (595,456 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\crypt32.dll
msvcrt	7.0.3790.3959 (srv03_sp2_rtm.070216-1710)	340.50 KB (348,672 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\msvcrt.dll
user32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	570.00 KB (583,680 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\user32.dll
gdi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	276.00 KB (282,624 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\gdi32.dll
msasn1	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	56.00 KB (57,344 bytes)	4/15/2008	Microsoft Corporation	c:\windows\system32\msasn1.dll

```

nddeapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
17.50 KB (17,920 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\nddeapi.dll
profmap 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
22.00 KB (22,528 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\profmap.dll
netapi32 5.2.3790.4392 (srv03_sp2_gdr.081016-1620)
337.00 KB (345,088 bytes) 12/2/2008
4:17 PM Microsoft Corporation
c:\windows\system32\netapi32.dll
userenv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
762.50 KB (780,800 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\userenv.dll
psapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
20.00 KB (20,480 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\psapi.dll
regapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
55.00 KB (56,320 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\regapi.dll
setupapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.02 MB (1,069,568 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\setupapi.dll
version 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
18.00 KB (18,432 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\version.dll
winsta 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
55.00 KB (56,320 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\winsta.dll
ws2_32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
81.50 KB (83,456 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\ws2_32.dll
ws2help 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
19.00 KB (19,456 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\ws2help.dll
msgina 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.15 MB (1,208,320 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\msgina.dll
shsvcs 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
132.00 KB (135,168 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\shsvcs.dll
shlwapi 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
312.50 KB (320,000 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\shlwapi.dll
sfc 5.2.3790.0 (srv03_rtm.030324-2048)
4.50 KB (4,608 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\sfc.dll
sfc_os 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
138.00 KB (141,312 bytes) 4/15/2008

```

```

12:38 PM Microsoft Corporation
c:\windows\system32\sfc_os.dll
wintrust 5.131.3790.3959 (srv03_sp2_rtm.070216-1710)
162.00 KB (165,888 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wintrust.dll
imagehlp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
144.50 KB (147,968 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\imagehlp.dll
ole32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.21 MB (1,267,200 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\ole32.dll
comctl32 6.0 (srv03_sp2_rtm.070216-1710)
1.00 MB (1,051,648 bytes) 2/18/2007
12:01 AM Microsoft Corporation
c:\windows\winsxs\x86_microsoft.windows.com
mon-controls_6595b64144ccfd1f_6.0.3790.3959_x-
ww_d8713e55\comctl32.dll
winscard 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
97.00 KB (99,328 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\winscard.dll
wtsapi32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
19.00 KB (19,456 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wtsapi32.dll
sxs 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
744.50 KB (762,368 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\sxs.dll
winmm 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
170.00 KB (174,080 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\winmm.dll
shell32 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
7.97 MB (8,359,936 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\shell32.dll
rsaenh 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
208.34 KB (213,336 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\rsaenh.dll
wldap32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
175.50 KB (179,712 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wldap32.dll
csddl 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
99.50 KB (101,888 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\csddl.dll
dimentfy 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
19.00 KB (19,456 bytes) 4/15/2008
12:42 PM Microsoft Corporation
c:\windows\system32\dimentfy.dll
wlnotify 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
94.50 KB (96,768 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wlnotify.dll
winspool 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
147.00 KB (150,528 bytes) 11/30/2005

```


6:00 AM Microsoft Corporation
 c:\windows\system32\winspool.drv
 mpr 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 56.50 KB (57,856 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\mpr.dll
 oleaut32 5.2.3790.3959 540.00 KB (552,960
 bytes) 11/30/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\oleaut32.dll
 comctl32 5.82 (srv03_sp2_rtm.070216-1710)
 585.00 KB (599,040 bytes) 2/18/2007
 12:01 AM Microsoft Corporation
 c:\windows\winsxs\x86_microsoft.windows.com
 mon-controls_6595b64144ccfldf_5.82.3790.3959_x-
 ww_78fcf8d0\comctl32.dll
 uxtheme 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
 202.00 KB (206,848 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\uxtheme.dll
 clbcatq 2001.12.4720.3959 (srv03_sp2_rtm.070216-
 1710) 499.00 KB (510,976 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\clbcatq.dll
 comres 2001.12.4720.3959 (srv03_sp2_rtm.070216-
 1710) 778.50 KB (797,184 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\comres.dll
 wbemprox 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 20.50 KB (20,992 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemprox.dll
 wbemcomn 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 220.50 KB (225,792 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemcomn.dll
 xpsp2res 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 2.76 MB (2,897,920 bytes) 4/15/2008
 12:41 PM Microsoft Corporation
 c:\windows\system32\xpsp2res.dll
 wbemsvc 5.2.3790.0 (srv03_rtm.030324-2048)
 42.50 KB (43,520 bytes) 8/7/2007 2:50
 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemsvc.dll
 fastprox 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 471.50 KB (482,816 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\fastprox.dll
 msvcpc60 7.0.3790.3959 (srv03_sp2_rtm.070216-1710)
 393.50 KB (402,944 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\msvcpc60.dll
 ntdsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 70.00 KB (71,680 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\ntdsapi.dll
 dnsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 156.50 KB (160,256 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\dnsapi.dll
 services 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 108.50 KB (111,104 bytes) 11/30/2005

6:00 AM Microsoft Corporation
 c:\windows\system32\services.exe
 scesrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 327.00 KB (334,848 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\scesrv.dll
 authz 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 69.00 KB (70,656 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\authz.dll
 umpnpgmr 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 125.00 KB (128,000 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\umpnpgmr.dll
 ncobjapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 36.00 KB (36,864 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\ncobjapi.dll
 eventlog 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 67.00 KB (68,608 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\eventlog.dll
 lsass 5.2.3790.0 (srv03_rtm.030324-2048)
 13.00 KB (13,312 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\lsass.exe
 lsasrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 796.00 KB (815,104 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\lsasrv.dll
 samsrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 444.00 KB (454,656 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\samsrv.dll
 cryptdll 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 32.50 KB (33,280 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\cryptdll.dll
 samlib 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 46.00 KB (47,104 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\samlib.dll
 msprive 5.2.3790.0 (srv03_rtm.030324-2048)
 46.50 KB (47,616 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\msprive.dll
 kerberos 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 342.50 KB (350,720 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\kerberos.dll
 msvl_0 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 140.00 KB (143,360 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\msvl_0.dll
 iphlpapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 93.00 KB (95,232 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\iphlpapi.dll
 netlogon 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 420.50 KB (430,592 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\netlogon.dll

w32time 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 222.00 KB (227,328 bytes) 4/15/2008
 12:37 PM Microsoft Corporation
 c:\windows\system32\w32time.dll
 schannel 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 143.50 KB (146,944 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\schannel.dll
 wdigest 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 74.50 KB (76,288 bytes) 4/15/2008
 12:37 PM Microsoft Corporation
 c:\windows\system32\wdigest.dll
 rassfm 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 23.00 KB (23,552 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\rassfm.dll
 kdcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 214.50 KB (219,648 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\kdcsvc.dll
 ntdsa 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 1.45 MB (1,522,176 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\ntdsa.dll
 ntdsatq 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 29.50 KB (30,208 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\ntdsatq.dll
 mssock 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 250.00 KB (256,000 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\mssock.dll
 esent 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 1,020.00 KB (1,044,480 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\esent.dll
 scecli 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 184.50 KB (188,928 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\scecli.dll
 ws03res 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 793.50 KB (812,544 bytes) 4/15/2008
 12:41 PM Microsoft Corporation
 c:\windows\system32\ws03res.dll
 pstorsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 24.50 KB (25,088 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\pstorsvc.dll
 psbase 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 84.00 KB (86,016 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\psbase.dll
 hnetcfg 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 347.00 KB (355,328 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\hnetcfg.dll
 wshtcpip 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 18.50 KB (18,944 bytes) 4/15/2008
 12:37 PM Microsoft Corporation
 c:\windows\system32\wshtcpip.dll
 w3ssl 6.0.3790.0 (srv03_rtm.030324-2048)
 15.00 KB (15,360 bytes) 11/30/2005

6:00 AM Microsoft Corporation
 c:\windows\system32\wsssl.dll
 strmfilt 6.0.3790.3959 (srv03_sp2_rtm.070216-1710)
 84.00 KB (86,016 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\strmfilt.dll
 httpapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 24.00 KB (24,576 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\httpapi.dll
 dssenh 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 143.84 KB (147,288 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\dssenh.dll
 svchost 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 14.50 KB (14,848 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\svchost.exe
 rpcss 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 470.50 KB (481,792 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\rpcss.dll
 ntmarta 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 119.00 KB (121,856 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\ntmarta.dll
 schedsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 197.50 KB (202,240 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\schedsvc.dll
 msidle 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
 6.50 KB (6,656 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\msidle.dll
 audiosrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 40.50 KB (41,472 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\audiosrv.dll
 wkssvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 130.00 KB (133,120 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\wkssvc.dll
 wiarpc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 32.50 KB (33,280 bytes) 4/15/2008
 12:37 PM Microsoft Corporation
 c:\windows\system32\wiarpc.dll
 aelupsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 26.00 KB (26,624 bytes) 4/15/2008
 12:42 PM Microsoft Corporation
 c:\windows\system32\aelupsvc.dll
 apphelp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 145.50 KB (148,992 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\apphelp.dll
 cryptsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 55.00 KB (56,320 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\cryptsvc.dll
 certcli 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 228.50 KB (233,984 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\certcli.dll

atl 3.05.2283 83.00 KB (84,992 bytes)
 11/30/2005 6:00 AM Microsoft Corporation
 c:\windows\system32\atl.dll
 vssapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 548.50 KB (561,664 bytes) 4/15/2008
 12:37 PM Microsoft Corporation
 c:\windows\system32\vssapi.dll
 dmserver 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 25.50 KB (26,112 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\dmserver.dll
 es 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)
 233.00 KB (238,592 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\es.dll
 pchsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 39.00 KB (39,936 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\pchsvc
 .dll
 srvsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 93.00 KB (95,232 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\srvsvc.dll
 seclogon 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 18.00 KB (18,432 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\seclogon.dll
 sens 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 36.50 KB (37,376 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\sens.dll
 trkwks 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 84.50 KB (86,528 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\trkwks.dll
 wmisvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 140.00 KB (143,360 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wmisvc.dll
 comsvcs 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)
 1.24 MB (1,295,872 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\comsvcs.dll
 browser 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 76.50 KB (78,336 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\browser.dll
 wbemcore 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 498.50 KB (510,464 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemcore.dll
 esscli 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 250.00 KB (256,000 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\esscli.dll
 wmiutils 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 93.50 KB (95,744 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wmiutils.dll
 repdrvfs 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 172.50 KB (176,640 bytes) 4/15/2008

12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\repdrvfs.dll
 wmiprvsd 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 404.00 KB (413,696 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wmiprvsd.dll
 wbemess 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 271.50 KB (278,016 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\wbemess.dll
 netrap 5.2.3790.0 (srv03_rtm.030324-2048)
 11.50 KB (11,776 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\netrap.dll
 ncprov 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 46.50 KB (47,616 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\wbem\ncprov.dll
 netman 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 257.50 KB (263,680 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\netman.dll
 netshell 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 1.73 MB (1,809,920 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\netshell.dll
 rtutils 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 34.00 KB (34,816 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\rtutils.dll
 credui 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 162.00 KB (165,888 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\credui.dll
 clusapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 59.50 KB (60,928 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\clusapi.dll
 mprapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 88.50 KB (90,624 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\mprapi.dll
 activeds 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 195.50 KB (200,192 bytes) 4/15/2008
 12:39 PM Microsoft Corporation
 c:\windows\system32\activeds.dll
 adslrpc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 149.50 KB (153,088 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\adslrpc.dll
 rasapi32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 240.50 KB (246,272 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rasapi32.dll
 rasman 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 61.50 KB (62,976 bytes) 11/30/2005
 6:00 AM Microsoft Corporation
 c:\windows\system32\rasman.dll
 tapi32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 179.50 KB (183,808 bytes) 4/15/2008
 12:38 PM Microsoft Corporation
 c:\windows\system32\tapi32.dll

wzcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 478.00 KB (489,472 bytes) 4/15/2008
Microsoft Corporation
12:37 PM c:\windows\system32\wzcsvc.dll
wmi 5.2.3790.0 (srv03_rtm.030324-2048) 6.50 KB (6,656 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wmi.dll
dhcpcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 114.50 KB (117,248 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\dhcpcsvc.dll
wininet 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 655.00 KB (670,720 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wininet.dll
wzcsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 42.00 KB (43,008 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\wzcsapi.dll
netcfgx 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 763.00 KB (781,312 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\netcfgx.dll
winipsec 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 36.00 KB (36,864 bytes) 4/15/2008
12:37 PM Microsoft Corporation
c:\windows\system32\winipsec.dll
rasmans 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 177.50 KB (181,760 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\rasmans.dll
rasdlg 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 663.00 KB (678,912 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\rasdlg.dll
rasadhlp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 7.50 KB (7,680 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\rasadhlp.dll
ntlsapi 5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB (8,192 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\ntlsapi.dll
wbemcons 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 45.50 KB (46,592 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\wbem\wbemcons.dll
aclient 6.9.164 5.10 MB (5,349,452 bytes) 8/13/2007 3:16 PM Altiris, Inc.
c:\program files\altiris\aclient\aclient.exe
comdlg32 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 267.00 KB (273,408 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\comdlg32.dll
wsoc32 5.2.3790.0 (srv03_rtm.030324-2048) 22.00 KB (22,528 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\wsoc32.dll
riched32 5.2.3790.0 (srv03_rtm.030324-2048) 3.50 KB (3,584 bytes) 11/30/2005

6:00 AM Microsoft Corporation
c:\windows\system32\riched32.dll
riched20 5.31.23.1225 433.00 KB (443,392 bytes) 4/15/2008 12:38 PM Microsoft Corporation
c:\windows\system32\riched20.dll
ersvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 24.00 KB (24,576 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\ersvc.dll
inetinfo 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 14.00 KB (14,336 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo.exe
iisutil 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 164.00 KB (167,936 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\iisutil.dll
rpcref 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 4.00 KB (4,096 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\rpcref.dll
iisrt1 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 138.50 KB (141,824 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\iisrt1.dll
iisadmin 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 21.00 KB (21,504 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\iisadmin.dll
coadmin 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 62.50 KB (64,000 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\coadmin.dll
admwprox 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 47.00 KB (48,128 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\admwprox.dll
iiscfg 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 1.08 MB (1,133,056 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\iiscfg.dll
metadata 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 229.00 KB (234,496 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\metadata.dll
msxml3 8.80.1185.0 1.08 MB (1,131,520 bytes) 4/15/2008 12:38 PM Microsoft Corporation
c:\windows\system32\msxml3.dll
svcxext 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 43.50 KB (44,544 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\svcxext.dll
security 5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\security.dll
iismap 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 58.50 KB (59,904 bytes) 4/15/2008

12:38 PM Microsoft Corporation
c:\windows\system32\iismap.dll
wamreg 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 54.50 KB (55,808 bytes) 4/15/2008
12:39 PM Microsoft Corporation
c:\windows\system32\inetinfo\wamreg.dll
sysdown 1.1.0.0 built by: builderv 6.50 KB (6,656 bytes) 8/13/2007 1:52 PM Hewlett-Packard Company
c:\windows\system32\sysdown.exe
iisw3adm 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 211.50 KB (216,576 bytes) 10/18/2007
3:19 PM Microsoft Corporation
c:\windows\system32\inetinfo\iisw3adm.dll
w3cache 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 19.00 KB (19,456 bytes) 10/18/2007
3:19 PM Microsoft Corporation
c:\windows\system32\inetinfo\w3cache.dll
w3tp 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 13.00 KB (13,312 bytes) 10/18/2007
3:19 PM Microsoft Corporation
c:\windows\system32\inetinfo\w3tp.dll
lonsint 6.0.3790.3959 (srv03_sp2_rtm.070216-1710) 13.00 KB (13,312 bytes) 10/18/2007
3:19 PM Microsoft Corporation
c:\windows\system32\inetinfo\lonsint.dll
termsrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 240.00 KB (245,760 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\termsrv.dll
icaapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 12.50 KB (12,800 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\icaapi.dll
mtlsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 117.00 KB (119,808 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\mtlsapi.dll
rdpwsx 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 102.13 KB (104,584 bytes) 4/15/2008
12:38 PM Microsoft Corporation
c:\windows\system32\rdpwsx.dll
msinfo32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 42.00 KB (43,008 bytes) 8/7/2007 2:52 PM
Microsoft Corporation
c:\program files\common files\microsoft shared\msinfo\msinfo32.exe
mfc42u 6.06.8063.0 1.11 MB (1,163,776 bytes) 11/30/2005 6:00 AM Microsoft Corporation
c:\windows\system32\mfc42u.dll
odbc32 3.526.3959.0 (srv03_sp2_rtm.070216-1710) 240.00 KB (245,760 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\odbc32.dll
odbcint 3.526.3959.0 (srv03_sp2_rtm.070216-1710) 92.00 KB (94,208 bytes) 11/30/2005
6:00 AM Microsoft Corporation
c:\windows\system32\odbcint.dll
msinfo 5.2.3790.3959 (srv03_sp2_rtm.070216-1710) 376.00 KB (385,024 bytes) 8/7/2007 2:52 PM
Microsoft Corporation

```

c:\windows\pchealth\helpctr\binaries\msinfo
.dll
[Services]
Display Name      Name      State      Start Mode
Service Type     Path      Error Control
Start Name       Tag ID
Altiris Client Service  AClient  Running
Auto            Own Process  c:\program
files\altiris\aclient\aclient.exe -service
Normal LocalSystem 0
Application Experience Lookup Service AeLookupSvc
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Alerter Alerter Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Application Layer Gateway Service ALG
Stopped Manual Own Process
c:\windows\system32\alg.exe Normal NT
AUTHORITY\LocalService 0
Application Management AppMgmt Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
ASP.NET State Service aspnet_state
Stopped Manual Own Process
c:\windows\microsoft.net\framework\v2.0.507
27\aspnet_state.exe Normal NT
AUTHORITY\NetworkService 0
Windows Audio AudioSrv Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Background Intelligent Transfer Service BITS
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Computer Browser Browser Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Indexing Service CiSvc Stopped Disabled
Share Process
c:\windows\system32\cisvc.exe Normal
LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
c:\windows\system32\clipsrv.exe
Normal LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_X86
clr_optimization_v2.0.50727_32
Stopped Manual Own Process
c:\windows\microsoft.net\framework\v2.0.507
27\mscorsvw.exe Ignore LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dllhost.exe

```

```

/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}
Normal LocalSystem 0
Cryptographic Services CryptSvc Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
DCOM Server Process Launcher DcomLaunch
Running Auto Share Process
c:\windows\system32\svchost.exe -k
dcomlaunch Normal LocalSystem 0
Distributed File System Dfs Stopped
Manual Own Process
c:\windows\system32\dfsrv.exe
Normal LocalSystem 0
DHCP Client Dhcp Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service
dmdadmin Stopped Manual Share Process
c:\windows\system32\dmdadmin.exe /com
Normal LocalSystem 0
Logical Disk Manager dmserver Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
DNS Client Dnscache Running Auto
Share Process
c:\windows\system32\svchost.exe -k
networkservice Normal NT
AUTHORITY\NetworkService 0
Error Reporting Service ERSvc Running
Auto Share Process
c:\windows\system32\svchost.exe -k winerr
Ignore LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\windows\system32\services.exe
Normal LocalSystem 0
COM+ Event System EventSystem Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Help and Support helpsvc Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Human Interface Device Access HidServ Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
HTTP SSL HTTPFilter Running Manual
Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
IIS Admin Service IISADMIN Running Auto
Share Process
c:\windows\system32\inetlib\inetinfo.exe
Normal LocalSystem 0
IMAPI CD-Burning COM Service ImapiService
Stopped Disabled Own Process

```

```

c:\windows\system32\imapi.exe Normal
LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own
Process c:\windows\system32\ismsserv.exe
Normal LocalSystem 0
Kerberos Key Distribution Center kdc
Stopped Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Server lanmanserver Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Workstation lanmanworkstation Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
License Logging LicenseService Stopped
Disabled Own Process
c:\windows\system32\llssrv.exe
Normal NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc
Stopped Disabled Own Process
c:\windows\system32\mnmsrvc.exe
Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC
Running Auto Own Process
c:\windows\system32\msdtc.exe Normal NT
AUTHORITY\NetworkService 0
Windows Installer MSIServer Stopped Manual
Share Process
c:\windows\system32\msiexec.exe /v
Normal LocalSystem 0
Visual Studio 2005 Remote Debugger msvsmon80
Stopped Disabled Own Process
"c:\program files\microsoft visual studio
8\common7\ide\remote debugger\x86\msvsmon.exe"
/service msvsmon80 Ignore LocalSystem 0
Network DDE NetDDE Stopped Disabled
Share Process
c:\windows\system32\netdde.exe
Normal LocalSystem 0
Network DDE DSDM NetDDEdsdm Stopped
Disabled Share Process
c:\windows\system32\netdde.exe
Normal LocalSystem 0
Net Logon Netlogon Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Network Connections Netman Running Manual
Share Process

```

```

c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Network Location Awareness (NLA) Nla
Running Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
File Replication NtFrfs Stopped Manual Own
Process c:\windows\system32\ntfrfs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp
Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Removable Storage NtmsSvc Stopped Manual
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Office Source Engine ose Stopped
Manual Own Process "c:\program
files\common files\microsoft shared\source
engine\ose.exe" Normal LocalSystem 0
Plug and Play PlugPlay Running Auto
Share Process
c:\windows\system32\services.exe
Normal LocalSystem 0
IPSEC Services PolicyAgent Stopped
Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Protected Storage ProtectedStorage Running
Auto Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Remote Access Auto Connection Manager RasAuto
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Access Connection Manager RasMan
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr
Stopped Manual Own Process
c:\windows\system32\sessmgr.exe
Normal LocalSystem 0
Routing and Remote Access RemoteAccess
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Remote Registry RemoteRegistry Running
Auto Share Process
c:\windows\system32\svchost.exe -k regsvc
Normal NT AUTHORITY\LocalService 0
Remote Command Service RMSYS Running
Auto Own Process "c:\program
files\benchmark\rsys.exe" Normal
.\Administrator 0
Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
c:\windows\system32\locator.exe

```

```

Normal NT AUTHORITY\NetworkService 0
Remote Procedure Call (RPC) RpcSs Running
Auto Share Process
c:\windows\system32\svchost.exe -k rpcss
Normal NT AUTHORITY\NetworkService 0
Resultant Set of Policy Provider RSoPProv
Stopped Manual Share Process
c:\windows\system32\rsopprov.exe
Normal LocalSystem 0
Special Administration Console Helper sacsrv
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Security Accounts Manager SamSs Running
Auto Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Smart Card SCardSvr Stopped Manual
Share Process
c:\windows\system32\scardsvr.exe
Ignore NT AUTHORITY\LocalService 0
Task Scheduler Schedule Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Secondary Logon seclogon Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
System Event Notification SENS Running
Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Windows Firewall/Internet Connection Sharing (ICS)
SharedAccess Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Shell Hardware Detection ShellHWDetection
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
Print Spooler Spooler Stopped Disabled Own
Process c:\windows\system32\spoolsv.exe
Normal LocalSystem 0
Windows Image Acquisition (WIA) stisvc
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k imgsvc
Normal NT AUTHORITY\LocalService 0
Microsoft Software Shadow Copy Provider swprv
Stopped Manual Own Process
c:\windows\system32\svchost.exe -k swprv
Normal LocalSystem 0
HP ProLiant System Shutdown Service sysdown
Running Auto Own Process
c:\windows\system32\sysdown.exe
Normal LocalSystem 0

```

```

Performance Logs and Alerts SysmonLog Stopped
Auto Own Process
c:\windows\system32\smlogsvc.exe
Normal NT AUTHORITY\NetworkService 0
Telephony TapiSrv Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k tapisrv
Normal LocalSystem 0
Terminal Services TermService Running
Manual Share Process
c:\windows\system32\svchost.exe -k termvcs
Normal LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process
c:\windows\system32\tlntsvr.exe
Normal NT AUTHORITY\LocalService 0
Distributed Link Tracking Server TrkSvr
Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Distributed Link Tracking Client TrkWks
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Terminal Services Session Directory Tssdis
Stopped Disabled Own Process
c:\windows\system32\tssdis.exe
Normal LocalSystem 0
Windows User Mode Driver Framework UMWdf
Stopped Manual Own Process
c:\windows\system32\wdfmgr.exe
Normal NT AUTHORITY\LocalService 0
Uninterruptible Power Supply UPS Stopped
Manual Own Process
c:\windows\system32\ups.exe Normal NT
AUTHORITY\LocalService 0
Virtual Disk Service vds Stopped
Manual Own Process
c:\windows\system32\vds.exe Normal
LocalSystem 0
Volume Shadow Copy VSS Stopped Manual Own
Process c:\windows\system32\vssvc.exe Normal
LocalSystem 0
Windows Time W32Time Running Auto
Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
World Wide Web Publishing Service W3SVC
Running Auto Share Process
c:\windows\system32\svchost.exe -k iissvcs
Normal LocalSystem 0
WebClient WebClient Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service
WinHttpAutoProxySvc Stopped Manual

```

```

Share Process
c:\windows\system32\svchost.exe -k
localservice Normal NT
AUTHORITY\LocalService 0
Windows Management Instrumentation winmgmt
Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs
Ignore LocalSystem 0
Portable Media Serial Number Service WmdmPmSN
Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped
Manual Own Process
c:\windows\system32\wbem\wmiaprv.exe
Normal LocalSystem 0
Automatic Updates wuauclt Stopped Disabled
Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Wireless Configuration WZCSVC Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0
Network Provisioning Service xmlprov Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs
Normal LocalSystem 0

```

[Program Groups]

Group Name	Name	User Name
Accessories	Default User:Accessories	
	Default User	
Accessories\Accessibility	Default	
User:Accessories\Accessibility		Default User
Accessories\Entertainment	Default	
User:Accessories\Entertainment		Default User
Startup	Default User:Startup	
		Default User
Accessories	All Users:Accessories	All Users
Accessories\Accessibility	All	All Users
Accessories\Communications	All	All Users
Accessories\Entertainment	All	All Users
Accessories\System Tools	All	All Users
Administrative Tools	All	All Users
HP System Tools	All Users:HP System Tools	All Users

```

HP System Tools\HP Array Diagnostic Utility All
Users:HP System Tools\HP Array Diagnostic Utility All
Users
Microsoft SQL Server 2005 All Users:Microsoft SQL
Server 2005 All Users
Microsoft SQL Server 2005\Analysis Services All
Users:Microsoft SQL Server 2005\Analysis Services All
Users
Microsoft SQL Server 2005\Configuration Tools All
Users:Microsoft SQL Server 2005\Configuration Tools
All Users
Microsoft SQL Server 2005\Documentation and Tutorials
All Users:Microsoft SQL Server
2005\Documentation and Tutorials All Users
Microsoft SQL Server 2005\Documentation and
Tutorials\Tutorials All Users:Microsoft SQL Server
2005\Documentation and Tutorials\Tutorials All
Users
Microsoft SQL Server 2005\Performance Tools All
Users:Microsoft SQL Server 2005\Performance Tools All
Users
Microsoft Visual Studio 2005 All Users:Microsoft
Visual Studio 2005 All Users
Microsoft Visual Studio 2005\Visual Studio Tools All
Users:Microsoft Visual Studio 2005\Visual Studio
Tools All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories
NT AUTHORITY\SYSTEM
Accessories\Accessibility NT
AUTHORITY\SYSTEM:Accessories\Accessibility NT
AUTHORITY\SYSTEM
Accessories\Entertainment NT
AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT
AUTHORITY\SYSTEM
Accessories CL121\Administrator:Accessories
CL121\Administrator
Accessories\Accessibility
CL121\Administrator:Accessories\Accessibili
ty
CL121\Administrator
Accessories\Entertainment
CL121\Administrator:Accessories\Entertainme
nt
CL121\Administrator
Administrative Tools
CL121\Administrator:Administrative Tools
CL121\Administrator
Benchcraft CL121\Administrator:Benchcraft
CL121\Administrator
Startup CL121\Administrator:Startup
CL121\Administrator

```

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	
desktop	desktop.ini	CL121\Administrator	
desktop	desktop.ini	.DEFAULT	Startup

```

desktop desktop.ini All Users Common
Startup
ACLntUsr c:\program
files\altiris\aclnt\aclntusr.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVers
ion\Run

```

[OLE Registration]

Object	Local Server	Not Available
Sound (OLE2)	sndrec32.exe	
Media Clip	mplay32.exe	
Video Clip	mplay32.exe /avi	
MIDI Sequence	mplay32.exe /mid	
Sound	Not Available	
Media Clip	Not Available	
WordPad Document	"%programfiles%\windows nt\accessories\wordpad.exe"	
Windows Media Services DRM Storage object		Not Available
Bitmap Image	mspaint.exe	

[Windows Error Reporting]

Time	Type	Details
------	------	---------

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Version	6.0.3790.3959
Build	63790.3959
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available

Cipher Strength	128-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path
actxprxy.dll	6.0.3790.3959	97 KB	2/17/2007 1:16:16 AM	C:\WINDOWS\system32 Microsoft Corporation
advpack.dll	6.0.3790.3959	98 KB	2/17/2007 1:16:46 AM	C:\WINDOWS\system32 Microsoft Corporation
asctrls.ocx	6.0.3790.0	90 KB	11/30/2005 6:00:00 AM	

C:\WINDOWS\system32 Microsoft Corporation

browseui.dll	6.0.3790.0	62 KB
11/30/2005 6:00:00 AM		
C:\WINDOWS\system32 Microsoft Corporation		
browseui.dll	6.0.3790.3959	1,009 KB
2/17/2007 1:22:54 AM		
C:\WINDOWS\system32 Microsoft Corporation		
cdfview.dll	6.0.3790.3959	148 KB
2/17/2007 1:23:26 AM		
C:\WINDOWS\system32 Microsoft Corporation		
comctl32.dll	5.82.3790.3959	585 KB
2/17/2007 1:31:40 AM		
C:\WINDOWS\system32 Microsoft Corporation		
dxtrans.dll	6.3.3790.3959	205 KB
2/17/2007 1:52:40 AM		
C:\WINDOWS\system32 Microsoft Corporation		
dxtmsft.dll	6.3.3790.3959	353 KB
2/17/2007 1:52:36 AM		
C:\WINDOWS\system32 Microsoft Corporation		
iecont.dll	<File Missing>	Not Available
Not Available Not Available Not Available		
iecontlc.dll	<File Missing>	Not Available
Not Available Not Available Not Available		
iedkcs32.dll	6.0.3790.3959	324 KB
2/17/2007 2:18:24 AM		
C:\WINDOWS\system32 Microsoft Corporation		
iepeers.dll	6.0.3790.3959	248 KB
2/17/2007 2:18:36 AM		
C:\WINDOWS\system32 Microsoft Corporation		
iesetup.dll	6.0.3790.3959	61 KB
2/17/2007 2:18:36 AM		
C:\WINDOWS\system32 Microsoft Corporation		
ieuinit.inf	Not Available	24 KB
2/17/2007 2:18:36 AM		
C:\WINDOWS\system32 Not Available		
iexplore.exe	6.0.3790.3959	92 KB
2/17/2007 2:18:36 AM		
C:\Program Files\Internet Explorer Microsoft Corporation		
imgutil.dll	6.0.3790.3959	38 KB
2/17/2007 2:19:34 AM		
C:\WINDOWS\system32 Microsoft Corporation		
inetcp1.cpl	6.0.3790.3959	361 KB
2/17/2007 2:19:44 AM		
C:\WINDOWS\system32 Microsoft Corporation		
inetcp1c.dll	6.0.3790.0	109 KB
11/30/2005 6:00:00 AM		

C:\WINDOWS\system32 Microsoft Corporation

inseng.dll	6.0.3790.3959	94 KB
2/17/2007 2:19:54 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mlang.dll	6.0.3790.3959	576 KB
2/17/2007 2:32:54 AM		
C:\WINDOWS\system32 Microsoft Corporation		
msencode.dll	2002.10.4.0	112 KB
11/30/2005 6:00:00 AM		
C:\WINDOWS\system32 ????		
mshta.exe	6.0.3790.3959	30 KB
2/17/2007 2:35:08 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mshtml.dll	6.0.3790.3959	3,058 KB
2/17/2007 2:35:20 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mshtml.tlb	6.0.3790.3959	1,320 KB
2/17/2007 2:35:20 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mshtml.dll	6.0.3790.3959	447 KB
2/17/2007 2:35:22 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mshtml.dll	6.0.3790.3959	56 KB
2/17/2007 2:35:24 AM		
C:\WINDOWS\system32 Microsoft Corporation		
msident.dll	6.0.3790.3959	48 KB
2/17/2007 2:35:30 AM		
C:\WINDOWS\system32 Microsoft Corporation		
msidntld.dll	6.0.3790.0	15 KB
11/30/2005 6:00:00 AM		
C:\WINDOWS\system32 Microsoft Corporation		
msieftp.dll	6.0.3790.3959	244 KB
2/17/2007 2:35:30 AM		
C:\WINDOWS\system32 Microsoft Corporation		
msrating.dll	6.0.3790.3959	144 KB
2/17/2007 2:36:24 AM		
C:\WINDOWS\system32 Microsoft Corporation		
mstime.dll	6.0.3790.3959	525 KB
2/17/2007 2:36:40 AM		
C:\WINDOWS\system32 Microsoft Corporation		
occache.dll	6.0.3790.3959	94 KB
2/17/2007 2:42:52 AM		
C:\WINDOWS\system32 Microsoft Corporation		
proctexe.ocx	6.3.3790.3959	83 KB
2/17/2007 2:52:42 AM		
C:\WINDOWS\system32 Intel Corporation		
sendmail.dll	6.0.3790.3959	56 KB
2/17/2007 2:58:56 AM		
C:\WINDOWS\system32 Microsoft Corporation		

shdoclc.dll	6.0.3790.0	589 KB
11/30/2005 6:00:00 AM		
C:\WINDOWS\system32 Microsoft Corporation		
shdocvw.dll	6.0.3790.3959	1,473 KB
2/17/2007 2:59:20 AM		
C:\WINDOWS\system32 Microsoft Corporation		
shfolder.dll	6.0.3790.3959	25 KB
2/17/2007 2:59:28 AM		
C:\WINDOWS\system32 Microsoft Corporation		
shlwapi.dll	6.0.3790.3959	313 KB
2/17/2007 2:59:42 AM		
C:\WINDOWS\system32 Microsoft Corporation		
tdc.ocx	1.3.0.3130	58 KB
11/30/2005 6:00:00 AM		
C:\WINDOWS\system32 Microsoft Corporation		
url.dll	6.0.3790.3959	37 KB
2/17/2007 3:07:34 AM		
C:\WINDOWS\system32 Microsoft Corporation		
urlmon.dll	6.0.3790.3959	682 KB
2/17/2007 3:07:36 AM		
C:\WINDOWS\system32 Microsoft Corporation		
webcheck.dll	6.0.3790.3959	271 KB
2/17/2007 3:08:42 AM		
C:\WINDOWS\system32 Microsoft Corporation		
wininet.dll	6.0.3790.3959	655 KB
2/17/2007 3:09:04 AM		
C:\WINDOWS\system32 Microsoft Corporation		
[Connectivity]		
Item	Value	
Connection Preference	Never dial	
LAN Settings		
AutoConfigProxy	wininet.dll	
AutoProxyDetectMode	Disabled	
AutoConfigURL		
Proxy	Disabled	
ProxyServer		
ProxyOverride		
[Cache]		
[Following are sub-categories of this main category]		
[Summary]		
Item	Value	
Page Refresh Type	Automatic	
Temporary Internet Files Folder	C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files	
Total Disk Space	Not Available	
Available Disk Space	Not Available	

Maximum Cache Size Not Available
Available Cache Size Not Available

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category
]

[Summary]

Item	Value
Content Advisor	Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone	Security Level
My Computer	Custom
Local intranet	Custom
Trusted sites	Custom
Internet	Custom
Restricted sites	Custom

Microsoft COM Component Configuration Parameters

The component services tool in Windows 2008 was used to change the queue settings for the TPCC COM+ queue components. All tpcc queue components were set to enable object pooling, object construction, just in time activation, and component supports events and statistics. The construction string was Server = myserver; UID= sa; pwd=; DATABASE= tpcc; The single

queue TpcAllTxn object was used, with the Min and Max both being set to 86 queues. Delivery threads were set under the TPCC key in the registry.

Internet Information Server Registry Parameters

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo\Parameters
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:41 PM

Value 0
Name: ListenBackLog
Type: REG_DWORD
Data: 0x19

Value 1
Name: PoolThreadLimit
Type: REG_DWORD
Data: 0xff4

Value 2
Name: MaxPoolThreads
Type: REG_DWORD
Data: 0x7fa

Value 3
Name: ThreadTimeout
Type: REG_DWORD
Data: 0x15180

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
InetInfo\Performance
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM

Value 0
Name: Library
Type: REG_SZ
Data: infoctrs.dll

Value 1
Name: Open
Type: REG_SZ
Data: OpenINFOPerformanceData

Value 2
Name: Close
Type: REG_SZ
Data: CloseINFOPerformanceData

Value 3
Name: Collect
Type: REG_SZ
Data: CollectINFOPerformanceData

Value 4
Name: PerfIniFile
Type: REG_SZ
Data: infoctrs.ini

Value 5
Name: Last Counter
Type: REG_DWORD
Data: 0xc30

Value 6
Name: Last Help
Type: REG_DWORD
Data: 0xc31

Value 7
Name: First Counter
Type: REG_DWORD
Data: 0xbf0

Value 8
Name: First Help
Type: REG_DWORD
Data: 0xbf1

Value 9
Name: Object List
Type: REG_SZ
Data: 3056

Value 10
Name: Library Validation Code
Type: REG_BINARY

00000000 00 b3 47 24 c4 11 c8 01 - 00 20 00 00 00
00 00 00 .'G\$Ä.È.

World Wide Web Service Registry Parameters


```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC
Class Name: <NO CLASS>
Last Write Time: 11/4/2009 - 2:51 PM
Value 0
  Name: Type
  Type: REG_DWORD
  Data: 0x20
Value 1
  Name: Start
  Type: REG_DWORD
  Data: 0x2
Value 2
  Name: ErrorControl
  Type: REG_DWORD
  Data: 0x1
Value 3
  Name: ImagePath
  Type: REG_EXPAND_SZ
  Data: %SystemRoot%\System32\svchost.exe
-k iissvcs
Value 4
  Name: DisplayName
  Type: REG_SZ
  Data: World Wide Web Publishing Service
Value 5
  Name: DependOnService
  Type: REG_MULTI_SZ
  Data: RPCSS
  HTTPFilter
  IISADMIN
Value 6
  Name: DependOnGroup
  Type: REG_MULTI_SZ
  Data:
Value 7
  Name: ObjectName
  Type: REG_SZ
  Data: LocalSystem
Value 8
  Name: Description
  Type: REG_SZ
  Data: Provides Web connectivity and
administration through the Internet Information
Services Manager
Value 9
  Name: FailureActions
  Type: REG_BINARY
  Data: 00000000 80 51 01 00 01 00 00 00 - 00 00 00 00 03
00 00 00 .Q.....

```

```

00000010 43 00 4c 00 01 00 00 00 - 01 00 00 00 01
00 00 00 C.L.....
01 00 00 00 01 00 00 00 - 01 00 00 00
.....
Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Parameters
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:41 PM
Value 0
  Name: MajorVersion
  Type: REG_DWORD
  Data: 0x6
Value 1
  Name: MinorVersion
  Type: REG_DWORD
  Data: 0
Value 2
  Name: InstallPath
  Type: REG_SZ
  Data: C:\WINDOWS\system32\inetsrv
Value 3
  Name: AccessDeniedMessage
  Type: REG_SZ
  Data: Error: Access is Denied.
Value 4
  Name: ServiceDll
  Type: REG_EXPAND_SZ
  Data: C:\WINDOWS\system32\inetsrv\iisw3adm.dll
Value 5
  Name: AcceptExOutstanding
  Type: REG_DWORD
  Data: 0x28
Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Parameters\ADCLaunch
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM
Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM
Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM

```

```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Performance
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM
Value 0
  Name: Library
  Type: REG_SZ
  Data: C:\WINDOWS\system32\inetsrv\w3ctrns.dll
Value 1
  Name: Open
  Type: REG_SZ
  Data: OpenW3PerformanceData
Value 2
  Name: Close
  Type: REG_SZ
  Data: CloseW3PerformanceData
Value 3
  Name: Collect
  Type: REG_SZ
  Data: CollectW3PerformanceData
Value 4
  Name: PerfIniFile
  Type: REG_SZ
  Data: w3ctrns.ini
Value 5
  Name: Last Counter
  Type: REG_DWORD
  Data: 0xd28
Value 6
  Name: Last Help
  Type: REG_DWORD
  Data: 0xd29
Value 7
  Name: First Counter
  Type: REG_DWORD
  Data: 0xc32
Value 8
  Name: First Help
  Type: REG_DWORD
  Data: 0xc33
Value 9
  Name: Object List
  Type: REG_SZ
  Data: 3122 3296
Value 10
  Name: Library Validation Code
  Type: REG_BINARY
  Data: 00000000 00 e0 78 25 c4 11 c8 01 - 00 5e 00 00 00
00 00 00 .ax%Ã.È..^.....

```

```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Security
Class Name: <NO CLASS>
Last Write Time: 10/18/2007 - 2:19 PM
Value 0
Name: Security
Type: REG_BINARY
Data:
00000000 01 00 14 80 b8 00 00 00 - c4 00 00 00 14
00 00 00 .....A.....
00000010 30 00 00 00 02 00 1c 00 - 01 00 00 00 02
80 14 00 0.....
00000020 ff 01 0f 00 01 01 00 00 - 00 00 00 01 00
00 00 00 y.....
00000030 02 00 88 00 06 00 00 00 - 00 00 14 00 fd
01 02 00 .....y...
00000040 01 01 00 00 00 00 00 05 - 12 00 00 00 00
00 18 00 .....
00000050 ff 01 0f 00 01 02 00 00 - 00 00 00 05 20
00 00 00 y.....
00000060 20 02 00 00 00 00 14 00 - 8d 01 02 00 01
01 00 00 .....
00000070 00 00 00 05 04 00 00 00 - 00 00 14 00 8d
01 02 00 .....
00000080 01 01 00 00 00 00 05 - 06 00 00 00 00
00 14 00 .....
00000090 00 01 00 00 01 01 00 00 - 00 00 00 05 0b
00 00 00 .....
000000a0 00 00 18 00 fd 01 02 00 - 01 02 00 00 00
00 00 05 .....y.....
000000b0 20 00 00 00 23 02 00 00 - 01 01 00 00 00
00 00 05 ...#.....
000000c0 12 00 00 00 01 01 00 00 - 00 00 00 05 12
00 00 00 .....

```

```

Key Name:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
W3SVC\Enum
Class Name: <NO CLASS>
Last Write Time: 11/4/2009 - 2:51 PM
Value 0
Name: 0
Type: REG_SZ
Data: Root\LEGACY_W3SVC\0000
Value 1
Name: Count
Type: REG_DWORD
Data: 0x1
Value 2
Name: NextInstance
Type: REG_DWORD
Data: 0x1

```

TPCC Application Registry Parameters

```

Key Name:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC
Class Name: <NO CLASS>
Last Write Time: 3/27/2009 - 9:54 AM
Value 0
Name: Path
Type: REG_SZ
Data: C:\Inetpub\wwwroot\
Value 1
Name: NumberOfDeliveryThreads
Type: REG_DWORD
Data: 0x12
Value 2
Name: MaxConnections
Type: REG_DWORD
Data: 0xc350
Value 3
Name: MaxPendingDeliveries
Type: REG_DWORD
Data: 0x7d0
Value 4
Name: DB_Protocol
Type: REG_SZ
Data: ODBC
Value 5
Name: TxnMonitor
Type: REG_SZ
Data: COM
Value 6
Name: DbServer
Type: REG_SZ
Data: hoops
Value 7
Name: DbName
Type: REG_SZ
Data: tpcc
Value 8
Name: DbUser
Type: REG_SZ
Data: sa
Value 9
Name: DbPassword

```

```

Type: REG_SZ
Data:
Value 10
Name: COM_SinglePool
Type: REG_SZ
Data: YES
Value 11
Name: CallNoDuplicatesNewOrder
Type: REG_DWORD
Data: 0x1
Value 12
Name: ConnectDelay
Type: REG_DWORD
Data: 0x1

```

Benchcraft Profile

```

Profile: hoops_53184_16c1
File Path: C:\Program
Files\BenchCraft\hoops_53184_16c1.xml
Version: 5
Number of Engines: 48
Name: d2
Description:
Directory: c:\blog\rte2.log
Machine: n31
Parameter Set: 2.2
Index: 160000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER53164609
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:
Name: d1
Description:
Directory: c:\blog\rte1.log
Machine: n31
Parameter Set: 2.2
Index: 750000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER44265281
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0

```

CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d3
Description:
Directory: c:\blog\рте3.log
Machine: n31
Parameter Set: 2.2
Index: 250000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3439676359
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d4
Description:
Directory: c:\blog\рте4.log
Machine: n32
Parameter Set: 2.2
Index: 300000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER4439706187
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d5
Description:
Directory: c:\blog\рте5.log
Machine: n32
Parameter Set: 2.2
Index: 400000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER5346413218
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d6
Description:
Directory: c:\blog\рте6.log
Machine: n32
Parameter Set: 2.2
Index: 500000000
Seed: 4678

Configured Users: 11080
Pipe Name: DRIVER62226046
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d7
Description:
Directory: c:\blog\рте7.log
Machine: n33
Parameter Set: 2.2
Index: 600000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER72289718
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d8
Description:
Directory: c:\blog\рте8.log
Machine: n33
Parameter Set: 2.2
Index: 220000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER82325578
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d9
Description:
Directory: c:\blog\рте9.log
Machine: n33
Parameter Set: 2.2
Index: 800000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER92360187
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d10

Description:
Directory: c:\blog\рте10.log
Machine: n34
Parameter Set: 2.2
Index: 900000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER102399796
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d11
Description:
Directory: c:\blog\рте11.log
Machine: n34
Parameter Set: 2.2
Index: 1000000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER1122682203
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d12
Description:
Directory: c:\blog\рте12.log
Machine: n34
Parameter Set: 2.2
Index: 1100000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER1222731546
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d13
Description:
Directory: c:\blog\рте13.log
Machine: n35
Parameter Set: 2.2
Index: 1200000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER13-1439076421
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0

Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d14
 Description:
 Directory: c:\blog\рте14.log
 Machine: n35
 Parameter Set: 2.2
 Index: 1300000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER14-1438943656
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d15
 Description:
 Directory: c:\blog\рте15.log
 Machine: n35
 Parameter Set: 2.2
 Index: 1400000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER15-1438852265
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d16
 Description:
 Directory: c:\blog\рте16.log
 Machine: n36
 Parameter Set: 2.2
 Index: 1500000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER16-1438790906
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d17
 Description:
 Directory: c:\blog\рте17.log
 Machine: n36
 Parameter Set: 2.2
 Index: 2150000000

Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER17-57150250
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d18
 Description:
 Directory: c:\blog\рте18.log
 Machine: n36
 Parameter Set: 2.2
 Index: 1700000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER18-57076468
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d19
 Description:
 Directory: c:\blog\рте19.log
 Machine: n37
 Parameter Set: 2.2
 Index: 1800000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER19-57030562
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d20
 Description:
 Directory: c:\blog\рте20.log
 Machine: n37
 Parameter Set: 2.2
 Index: 1900000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER20-56992625
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

Name: d21
 Description:
 Directory: c:\blog\рте21.log
 Machine: n37
 Parameter Set: 2.2
 Index: 27000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER2191781
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d22
 Description:
 Directory: c:\blog\рте22.log
 Machine: n38
 Parameter Set: 2.2
 Index: 2100000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER221814250
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d23
 Description:
 Directory: c:\blog\рте23.log
 Machine: n38
 Parameter Set: 2.2
 Index: 300000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER231877968
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d24
 Description:
 Directory: c:\blog\рте24.log
 Machine: n38
 Parameter Set: 2.2
 Index: 40000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER242206343
 Connect Rate: 10
 Start Rate: 0

Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d25
Description:
Directory: c:\blog\rte25.log
Machine: n39
Parameter Set: 2.2
Index: 50000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER252251500
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d26
Description:
Directory: c:\blog\rte26.log
Machine: n39
Parameter Set: 2.2
Index: 60000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER262289250
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d27
Description:
Directory: c:\blog\rte27.log
Machine: n39
Parameter Set: 2.2
Index: 70000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER272340437
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d28
Description:
Directory: c:\blog\rte28.log
Machine: n41
Parameter Set: 2.2

Index: 80000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER282382234
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d29
Description:
Directory: c:\blog\rte29.log
Machine: n41
Parameter Set: 2.2
Index: 90000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER292416328
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d30
Description:
Directory: c:\blog\rte30.log
Machine: n41
Parameter Set: 2.2
Index: 100000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER302463687
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d31
Description:
Directory: c:\blog\rte31.log
Machine: n42
Parameter Set: 2.2
Index: 25500000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3155814328
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d32
Description:
Directory: c:\blog\rte32.log
Machine: n42
Parameter Set: 2.2
Index: 35500000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3255892765
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d33
Description:
Directory: c:\blog\rte33.log
Machine: n42
Parameter Set: 2.2
Index: 45500000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3355948500
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Name: d34
Description:
Directory: c:\blog\rte34.log
Machine: n43
Parameter Set: 2.2
Index: 55500000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3455990593
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d35
Description:
Directory: c:\blog\rte35.log
Machine: n43
Parameter Set: 2.2
Index: 65500000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER3556027390
Connect Rate: 10

Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d36
 Description:
 Directory: c:\blog\rte36.log
 Machine: n43
 Parameter Set: 2.2
 Index: 75500000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER3656077062
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d37
 Description:
 Directory: c:\blog\rte37.log
 Machine: n27
 Parameter Set: 2.2
 Index: 2105000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER37766536203
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d38
 Description:
 Directory: c:\blog\rte38.log
 Machine: n27
 Parameter Set: 2.2
 Index: 2050000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER3876654375
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d39
 Description:
 Directory: c:\blog\rte39.log
 Machine: n27

Parameter Set: 2.2
 Index: 1905000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER39766760968
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d40
 Description:
 Directory: c:\blog\rte40.log
 Machine: n28
 Parameter Set: 2.2
 Index: 7050000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER40766820328
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d41
 Description:
 Directory: c:\blog\rte38.log
 Machine: n28
 Parameter Set: 2.2
 Index: 1805000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER41766909890
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d42
 Description:
 Directory: c:\blog\rte42.log
 Machine: n28
 Parameter Set: 2.2
 Index: 1705000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER42766941343
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2

Additional Options:

 Name: d43
 Description:
 Directory: c:\blog\rte43.log
 Machine: n29
 Parameter Set: 2.2
 Index: 1605000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER43766990906
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 0
 Additional Options:

 Name: d44
 Description:
 Directory: c:\blog\rte44.log
 Machine: n29
 Parameter Set: 2.2
 Index: 1505000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER44767023437
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 1
 Additional Options:

 Name: d45
 Description:
 Directory: c:\blog\rte45.log
 Machine: n29
 Parameter Set: 2.2
 Index: 1105000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER45767085000
 Connect Rate: 10
 Start Rate: 0
 Max. Concurrency: 0
 Concurrency Rate: 0
 CLIENT_NURAND: 25
 CPU: 2
 Additional Options:

 Name: d46
 Description:
 Directory: c:\blog\rte46.log
 Machine: n30
 Parameter Set: 2.2
 Index: 1050000000
 Seed: 4678
 Configured Users: 11080
 Pipe Name: DRIVER46767120687

Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 0
Additional Options:

Name: d47
Description:
Directory: c:\blog\rte47.log
Machine: n30
Parameter Set: 2.2
Index: 905000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER47767168296
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 1
Additional Options:

Name: d48
Description:
Directory: c:\blog\rte48.log
Machine: n30
Parameter Set: 2.2
Index: 805000000
Seed: 4678
Configured Users: 11080
Pipe Name: DRIVER48767212015
Connect Rate: 10
Start Rate: 0
Max. Concurrency: 0
Concurrency Rate: 0
CLIENT_NURAND: 25
CPU: 2
Additional Options:

Number of User groups: 48

Driver Engine: d1
IIS Server: cr121
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 1108
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d2
IIS Server: cr121
SQL Server: hoops
Database: tpcc

User: sa
Protocol: HTML
w_id Range: 1109 - 2216
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d3
IIS Server: cr121
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2217 - 3324
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d4
IIS Server: cr122
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3325 - 4432
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d5
IIS Server: cr122
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4433 - 5540
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d6
IIS Server: cr122
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5541 - 6648
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal

User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d7
IIS Server: cr123
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6649 - 7756
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d8
IIS Server: cr123
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7757 - 8864
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d9
IIS Server: cr123
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8865 - 9972
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d10
IIS Server: cr124
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 9973 - 11080
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d11
IIS Server: cr124

SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 11081 - 12188
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d12
IIS Server: cr124
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 12189 - 13296
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d13
IIS Server: cr125
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 13297 - 14404
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d14
IIS Server: cr125
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 14405 - 15512
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d15
IIS Server: cr125
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 15513 - 16620
w_id Min Warehouse: 1

w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d16
IIS Server: cr126
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 16621 - 17728
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d17
IIS Server: cr126
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 17729 - 18836
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d18
IIS Server: cr126
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 18837 - 19944
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d19
IIS Server: cr127
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 19945 - 21052
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d20
IIS Server: cr127
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 21053 - 22160
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d21
IIS Server: cr127
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 22161 - 23268
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d22
IIS Server: cr128
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 23269 - 24376
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d23
IIS Server: cr128
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 24377 - 25484
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d24
IIS Server: cr128
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML

w_id Range: 25485 - 26592
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d25
IIS Server: cr129
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 26593 - 27700
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d26
IIS Server: cr129
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 27701 - 28808
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d27
IIS Server: cr129
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 28809 - 29916
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d28
IIS Server: cr130
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 29917 - 31024
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1

Scale Down: No

Driver Engine: d29
IIS Server: cr130
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 31025 - 32132
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d30
IIS Server: cr130
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 32133 - 33240
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d31
IIS Server: cr131
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 33241 - 34348
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d32
IIS Server: cr131
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 34349 - 35456
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d33
IIS Server: cr131
SQL Server: hoops
Database: tpcc

User: sa
Protocol: HTML
w_id Range: 35457 - 36564
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d34
IIS Server: cr132
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 36565 - 37672
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d35
IIS Server: cr132
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 37673 - 38780
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d36
IIS Server: cr132
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 38781 - 39888
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal
User Count: 11080
District id: 1
Scale Down: No

Driver Engine: d37
IIS Server: cr77
SQL Server: hoops
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 39889 - 40996
w_id Min Warehouse: 1
w_id Max Warehouse: 53184
Scale: Normal

User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d38
 IIS Server: cr77
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 40997 - 42104
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d39
 IIS Server: cr77
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 42105 - 43212
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d40
 IIS Server: cr78
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 43213 - 44320
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d41
 IIS Server: cr78
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 44321 - 45428
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d42
 IIS Server: cr78

SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 45429 - 46536
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d43
 IIS Server: cr79
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 46537 - 47644
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d44
 IIS Server: cr79
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 47645 - 48752
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d45
 IIS Server: cr79
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 48753 - 49860
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d46
 IIS Server: cr80
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 49861 - 50968
 w_id Min Warehouse: 1

w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d47
 IIS Server: cr80
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 50969 - 52076
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Driver Engine: d48
 IIS Server: cr80
 SQL Server: hoops
 Database: tpcc
 User: sa
 Protocol: HTML
 w_id Range: 52077 - 53184
 w_id Min Warehouse: 1
 w_id Max Warehouse: 53184
 Scale: Normal
 User Count: 11080
 District id: 1
 Scale Down: No

Number of Parameter Sets: 69

-Default						
Default Parameter Set						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	10.00		
12.05		18.01	0.10	5.00	0.10	
			Payment	10.00		
12.05		3.01	0.10	5.00	0.10	
			Delivery	1.00		
5.05		2.01	0.10	5.00	0.10	
			Stock Level	1.00		
5.05		2.01	0.10	20.00	0.10	
			Order Status	1.00		
10.05		2.01	0.10	5.00	0.10	
Tuned Distribution						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.75		
12.05		18.01	0.10	5.00	0.10	

12.05	3.01	Payment	43.10
		0.10	5.00 0.10
5.05	2.01	Delivery	4.05
		0.10	5.00 0.10
5.05	2.01	Stock Level	4.05
		0.10	20.00 0.10
10.05	2.01	Order Status	4.05
		0.10	5.00 0.10

No Think

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
0.00	0.00	New Order	10.00		
		0.00	5.00 0.00		
0.00	0.00	Payment	10.00		
		0.00	5.00 0.00		
0.00	0.00	Delivery	1.00		
		0.00	5.00 0.00		
0.00	0.00	Stock Level	1.00		
		0.00	20.00 0.00		
0.00	0.00	Order Status	1.00		
		0.00	5.00 0.00		

95%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.00	18.01	New Order	44.75		
		0.10	5.00 0.10		
13.00	3.01	Payment	43.10		
		0.10	5.00 0.10		
6.00	2.01	Delivery	4.05		
		0.10	5.00 0.10		
6.00	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
11.00	2.01	Order Status	4.05		
		0.10	5.00 0.10		

90%

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
16.00	18.01	New Order	44.83		
		0.10	5.00 0.10		
16.00	3.01	Payment	43.05		
		0.10	5.00 0.10		
9.00	2.01	Delivery	4.04		
		0.10	5.00 0.10		
9.00	2.01	Stock Level	4.04		
		0.10	20.00 0.10		
14.00	2.01	Order Status	4.04		
		0.10	5.00 0.10		

3.0

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
36.15	0.00	New Order	44.75		
		0.10	5.00 0.10		
36.15	0.00	Payment	43.10		
		0.10	5.00 0.10		
15.15	0.00	Delivery	4.05		
		0.10	5.00 0.10		
15.15	0.00	Stock Level	4.05		
		0.10	20.00 0.10		
30.15	0.00	Order Status	4.05		
		0.10	5.00 0.10		

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
48.20	18.01	New Order	44.75		
		0.10	5.00 0.10		
48.20	3.01	Payment	43.10		
		0.10	5.00 0.10		
20.20	2.01	Delivery	4.05		
		0.10	5.00 0.10		
20.20	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
40.20	2.01	Order Status	4.05		
		0.10	5.00 0.10		

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
45.70	18.01	New Order	44.75		
		0.10	5.00 0.10		
45.70	3.01	Payment	43.10		
		0.10	5.00 0.10		
19.10	2.01	Delivery	4.05		
		0.10	5.00 0.10		
19.10	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
38.10	2.01	Order Status	4.05		
		0.10	5.00 0.10		

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
43.30	18.01	New Order	44.75		
		0.10	5.00 0.10		
43.30	3.01	Payment	43.10		
		0.10	5.00 0.10		
18.10	2.01	Delivery	4.05		
		0.10	5.00 0.10		

18.10	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
36.18	2.01	Order Status	4.05		
		0.10	5.00 0.10		

3.4

3.4 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
40.90	18.01	New Order	44.75		
		0.10	5.00 0.10		
40.90	3.01	Payment	43.10		
		0.10	5.00 0.10		
17.10	2.01	Delivery	4.05		
		0.10	5.00 0.10		
17.10	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
17.10	2.01	Order Status	4.05		
		0.10	5.00 0.10		

3.2

3.2 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
38.50	18.01	New Order	44.75		
		0.10	5.00 0.10		
38.50	3.01	Payment	43.10		
		0.10	5.00 0.10		
16.10	2.01	Delivery	4.05		
		0.10	5.00 0.10		
16.10	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
32.10	2.01	Order Status	4.05		
		0.10	5.00 0.10		

2.8

2.8 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
33.74	18.01	New Order	44.75		
		0.10	5.00 0.10		
33.74	3.01	Payment	43.10		
		0.10	5.00 0.10		
14.14	2.01	Delivery	4.05		
		0.10	5.00 0.10		
14.14	2.01	Stock Level	4.05		
		0.10	20.00 0.10		
28.14	2.01	Order Status	4.05		
		0.10	5.00 0.10		

2.6

2.6 tt

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

31.30	18.01		New Order	44.75		
			0.10	5.00	0.10	
31.30	3.01		Payment	43.10		
			0.10	5.00	0.10	
13.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
13.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
26.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.4			
			2.4 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
28.90	18.01		New Order	44.75		
			0.10	5.00	0.10	
28.90	3.01		Payment	43.10		
			0.10	5.00	0.10	
12.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
12.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
24.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.2			
			2.2 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
28.90	18.01		New Order	44.75		
			0.10	5.00	0.10	
28.90	3.01		Payment	43.10		
			0.10	5.00	0.10	
12.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
12.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
24.12	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			2.0			
			2.0 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
24.10	18.01		New Order	44.75		
			0.10	5.00	0.10	
24.10	3.01		Payment	43.10		
			0.10	5.00	0.10	
10.10	2.01		Delivery	4.05		
			0.10	5.00	0.10	
10.10	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
20.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	

			5.0			
			5.0 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
60.25	18.01		New Order	44.75		
			0.10	5.00	0.10	
60.25	3.01		Payment	43.10		
			0.10	5.00	0.10	
25.25	2.01		Delivery	4.05		
			0.10	5.00	0.10	
25.25	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
50.25	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			4.5			
			4.5 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
54.20	18.01		New Order	44.75		
			0.10	5.00	0.10	
54.20	3.01		Payment	43.10		
			0.10	5.00	0.10	
22.70	2.01		Delivery	4.05		
			0.10	5.00	0.10	
22.70	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
45.20	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			3.5			
			3.5 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
42.10	18.01		New Order	44.75		
			0.10	5.00	0.10	
42.10	3.01		Payment	43.10		
			0.10	5.00	0.10	
17.60	2.01		Delivery	4.05		
			0.10	5.00	0.10	
17.60	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
35.10	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			1.8			
			1.8 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
21.60	18.01		New Order	44.75		
			0.10	5.00	0.10	
21.60	3.01		Payment	43.10		
			0.10	5.00	0.10	

9.09	2.01		Delivery	4.05		
			0.10	5.00	0.10	
9.09	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
18.09	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			4.2			
			4.2 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
54.20	18.01		New Order	44.75		
			0.10	5.00	0.10	
54.20	3.01		Payment	43.10		
			0.10	5.00	0.10	
22.70	2.01		Delivery	4.05		
			0.10	5.00	0.10	
22.70	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
45.20	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			1.6			
			1.6 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
19.20	18.01		New Order	44.75		
			0.10	5.00	0.10	
19.20	3.01		Payment	43.10		
			0.10	5.00	0.10	
8.08	2.01		Delivery	4.05		
			0.10	5.00	0.10	
8.08	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
16.08	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			1.4			
			1.4 tt			
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
16.87	18.01		New Order	44.75		
			0.10	5.00	0.10	
16.87	3.01		Payment	43.10		
			0.10	5.00	0.10	
7.07	2.01		Delivery	4.05		
			0.10	5.00	0.10	
7.07	2.01		Stock Level	4.05		
			0.10	20.00	0.10	
14.07	2.01		Order Status	4.05		
			0.10	5.00	0.10	
			1.2			
			1.2 tt			
Key	RT	RT	Menu	Txn	Think	

Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
14.46	18.01		0.10	5.00	0.10
			Payment	43.05	
14.46	3.01		0.10	5.00	0.10
			Delivery	4.04	
6.06	2.01		0.10	5.00	0.10
			Stock Level	4.04	
6.06	2.01		0.10	20.00	0.10
			Order Status	4.04	
12.06	2.01		0.10	5.00	0.10
			3.5		
			3.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.75	
42.10	18.01		0.10	5.00	0.10
			Payment	43.10	
42.10	3.01		0.10	5.00	0.10
			Delivery	4.05	
17.60	2.01		0.10	5.00	0.10
			Stock Level	4.05	
17.60	2.01		0.10	20.00	0.10
			Order Status	4.05	
35.10	2.01		0.10	5.00	0.10
			1.9		
			1.9 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.75	
22.89	18.01		0.10	5.00	0.10
			Payment	43.10	
22.89	3.01		0.10	5.00	0.10
			Delivery	4.05	
9.59	2.01		0.10	5.00	0.10
			Stock Level	4.05	
9.59	2.01		0.10	20.00	0.10
			Order Status	4.05	
19.09	2.01		0.10	5.00	0.10
			1.1		
			1.1 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
13.25	18.01		0.10	5.00	0.10
			Payment	43.05	
13.25	3.01		0.10	5.00	0.10
			Delivery	4.04	
5.55	2.01		0.10	5.00	0.10
			Stock Level	4.04	
5.55	2.01		0.10	20.00	0.10

Time	Delay	Fence	Delay	Weight	Time
			New Order	44.92	
12.65	18.01		0.10	5.00	0.10
			Payment	43.01	
12.65	3.01		0.10	5.00	0.10
			Delivery	4.02	
5.30	2.01		0.10	5.00	0.10
			Stock Level	4.03	
5.30	2.01		0.10	20.00	0.10
			Order Status	4.02	
10.55	2.01		0.10	5.00	0.10
			1.09		
			1.09 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
13.13	18.01		0.10	5.00	0.10
			Payment	43.05	
13.13	3.01		0.10	5.00	0.10
			Delivery	4.04	
5.50	2.01		0.10	5.00	0.10
			Stock Level	4.04	
5.50	2.01		0.10	20.00	0.10
			Order Status	4.04	
10.95	2.01		0.10	5.00	0.10
			1.08		
			1.08 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
13.01	18.01		0.10	5.00	0.10
			Payment	43.05	
13.01	3.01		0.10	5.00	0.10
			Delivery	4.04	
5.45	2.01		0.10	5.00	0.10
			Stock Level	4.04	
5.45	2.01		0.10	20.00	0.10
			Order Status	4.04	
10.85	2.01		0.10	5.00	0.10
			1.07		
			1.07 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
12.89	18.01		0.10	5.00	0.10

Time	Delay	Fence	Delay	Weight	Time
			New Order	44.83	
12.89	3.01		0.10	5.00	0.10
			Payment	43.05	
5.40	2.01		0.10	5.00	0.10
			Stock Level	4.04	
5.40	2.01		0.10	20.00	0.10
			Order Status	4.04	
10.75	2.01		0.10	5.00	0.10
			1.06		
			1.06 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
12.77	18.01		0.10	5.00	0.10
			Payment	43.05	
12.77	3.01		0.10	5.00	0.10
			Delivery	4.04	
5.35	2.01		0.10	5.00	0.10
			Stock Level	4.04	
5.35	2.01		0.10	20.00	0.10
			Order Status	4.04	
10.65	2.01		0.10	5.00	0.10
			1.15		
			1.15 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.75	
13.85	18.01		0.10	5.00	0.10
			Payment	43.10	
13.85	3.01		0.10	5.00	0.10
			Delivery	4.05	
5.80	2.01		0.10	5.00	0.10
			Stock Level	4.05	
5.80	2.01		0.10	20.00	0.10
			Order Status	4.05	
11.55	2.01		0.10	5.00	0.10
			1.25		
			1.25 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay		
			New Order	44.83	
15.06	18.01		0.10	5.00	0.10
			Payment	43.05	
15.06	3.01		0.10	5.00	0.10
			Delivery	4.04	
6.31	2.01		0.10	5.00	0.10
			Stock Level	4.04	
6.31	2.01		0.10	20.00	0.10
			Order Status	4.04	
12.56	2.01		0.10	5.00	0.10
			1.3		
			1.3 tt		

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
15.66	18.01		New Order	44.83	
			0.10	5.00	0.10
15.66	3.01		Payment	43.05	
			0.10	5.00	0.10
6.56	2.01		Delivery	4.04	
			0.10	5.00	0.10
6.56	2.01		Stock Level	4.04	
			0.10	20.00	0.10
13.06	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.12		
			1.12 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
13.49	18.01		New Order	44.75	
			0.10	5.00	0.10
13.49	3.01		Payment	43.10	
			0.10	5.00	0.10
5.65	2.01		Delivery	4.05	
			0.10	5.00	0.10
5.65	2.01		Stock Level	4.05	
			0.10	20.00	0.10
11.25	2.01		Order Status	4.05	
			0.10	5.00	0.10
			1.18		
			1.18 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
14.21	18.01		New Order	44.75	
			0.10	5.00	0.10
14.21	3.01		Payment	43.10	
			0.10	5.00	0.10
5.95	2.01		Delivery	4.05	
			0.10	5.00	0.10
5.95	2.01		Stock Level	4.05	
			0.10	20.00	0.10
11.85	2.01		Order Status	4.05	
			0.10	5.00	0.10
			1.22		
			1.22 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
14.70	18.01		New Order	44.75	
			0.10	5.00	0.10
14.70	3.01		Payment	43.10	
			0.10	5.00	0.10
6.16	2.01		Delivery	4.05	
			0.10	5.00	0.10

6.16	2.01		Stock Level	4.05	
			0.10	20.00	0.10
12.26	2.01		Order Status	4.05	
			0.10	5.00	0.10
			1.28		
			1.28 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
15.42	18.01		New Order	44.75	
			0.10	5.00	0.10
15.42	3.01		Payment	43.10	
			0.10	5.00	0.10
6.46	2.01		Delivery	4.05	
			0.10	5.00	0.10
6.46	2.01		Stock Level	4.05	
			0.10	20.00	0.10
12.86	2.01		Order Status	4.05	
			0.10	5.00	0.10
			1.04		
			1.04 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.53	18.01		New Order	44.83	
			0.10	5.00	0.10
12.53	3.01		Payment	43.05	
			0.10	5.00	0.10
5.25	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.25	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.45	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.03		
			1.03 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.41	18.01		New Order	44.83	
			0.10	5.00	0.10
12.41	3.01		Payment	43.05	
			0.10	5.00	0.10
5.20	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.20	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.35	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.02		
			1.02 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time

12.29	18.01		New Order	44.83	
			0.10	5.00	0.10
12.29	3.01		Payment	43.05	
			0.10	5.00	0.10
5.15	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.15	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.25	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.01		
			1.01 tt		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01		New Order	44.83	
			0.10	5.00	0.10
12.17	3.01		Payment	43.05	
			0.10	5.00	0.10
5.10	2.01		Delivery	4.04	
			0.10	5.00	0.10
5.10	2.01		Stock Level	4.04	
			0.10	20.00	0.10
10.15	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.005_best		
			1.005 tt best		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.11	18.01		New Order	44.88	
			0.10	5.00	0.10
12.11	3.01		Payment	43.02	
			0.10	5.00	0.10
5.07	2.01		Delivery	4.03	
			0.10	5.00	0.10
5.07	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.10	2.01		Order Status	4.03	
			0.10	5.00	0.10
			1.001_best		
			1.001 tt best		
Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
12.06	18.01		New Order	44.91	
			0.10	5.00	0.10
12.06	3.01		Payment	43.04	
			0.10	5.00	0.10
5.06	2.01		Delivery	4.01	
			0.10	5.00	0.10
5.06	2.01		Stock Level	4.02	
			0.10	20.00	0.10
10.06	2.01		Order Status	4.02	
			0.10	5.00	0.10

1.03 better						
1.03 tt more aggressive						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.92		
12.41	18.01		0.10	5.00	0.10	
			Payment	43.01		
12.41	3.01		0.10	5.00	0.10	
			Delivery	4.02		
5.20	2.01		0.10	5.00	0.10	
			Stock Level	4.03		
5.20	2.01		0.10	20.00	0.10	
			Order Status	4.02		
10.35	2.01		0.10	5.00	0.10	

1.005 better						
1.005 tt more aggressive						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.90		
12.11	18.01		0.10	5.00	0.10	
			Payment	43.05		
12.11	3.01		0.10	5.00	0.10	
			Delivery	4.01		
5.07	2.01		0.10	5.00	0.10	
			Stock Level	4.03		
5.07	2.01		0.10	20.00	0.10	
			Order Status	4.01		
10.10	2.01		0.10	5.00	0.10	

1.02 better						
1.02 tt more aggressive						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.92		
12.29	18.01		0.10	5.00	0.10	
			Payment	43.01		
12.29	3.01		0.10	5.00	0.10	
			Delivery	4.02		
5.15	2.01		0.10	5.00	0.10	
			Stock Level	4.03		
5.15	2.01		0.10	20.00	0.10	
			Order Status	4.02		
10.25	2.01		0.10	5.00	0.10	

1.01 best						
1.01 tt best						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.90		
12.17	18.01		0.10	5.00	0.10	
			Payment	43.05		
12.17	3.01		0.10	5.00	0.10	

5.10	2.01		Delivery	4.01		
			0.10	5.00	0.10	
			Stock Level	4.03		
5.10	2.01		0.10	20.00	0.10	
			Order Status	4.01		
10.15	2.01		0.10	5.00	0.10	

1.02 best						
1.02 tt best						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.96		
12.29	18.01		0.00	5.00	0.00	
			Payment	43.00		
12.29	3.01		0.00	5.00	0.00	
			Delivery	4.00		
5.15	2.01		0.00	5.00	0.00	
			Stock Level	4.03		
5.15	2.01		0.00	20.00	0.00	
			Order Status	4.01		
10.25	2.01		0.00	5.00	0.00	

1.03 best						
1.03 tt best						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.96		
12.41	18.01		0.10	5.00	0.10	
			Payment	43.01		
12.41	3.01		0.10	5.00	0.10	
			Delivery	4.01		
5.20	2.01		0.10	5.00	0.10	
			Stock Level	4.01		
5.20	2.01		0.10	20.00	0.10	
			Order Status	4.01		
10.35	2.01		0.10	5.00	0.10	

5.5						
5.5 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
66.28	18.01		0.10	5.00	0.10	
			Payment	43.05		
66.28	3.01		0.10	5.00	0.10	
			Delivery	4.04		
27.77	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
27.77	2.01		0.10	20.00	0.10	
			Order Status	4.04		
55.27	2.01		0.10	5.00	0.10	

6.0						
6.0 tt						
Key	RT	RT	Menu	Txn	Think	
			New Order	44.83		
66.28	18.01		0.10	5.00	0.10	
			Payment	43.05		
66.28	3.01		0.10	5.00	0.10	
			Delivery	4.04		
27.77	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
27.77	2.01		0.10	20.00	0.10	
			Order Status	4.04		
55.27	2.01		0.10	5.00	0.10	

Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
72.30	18.01		0.10	5.00	0.10	
			Payment	43.05		
72.30	3.01		0.10	5.00	0.10	
			Delivery	4.04		
30.30	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
30.30	2.01		0.10	20.00	0.10	
			Order Status	4.04		
60.30	2.01		0.10	5.00	0.10	

6.5						
6.5 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
79.53	18.01		0.10	5.00	0.10	
			Payment	43.05		
79.53	3.01		0.10	5.00	0.10	
			Delivery	4.04		
33.33	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
33.33	2.01		0.10	20.00	0.10	
			Order Status	4.04		
66.33	2.01		0.10	5.00	0.10	

7.0						
7.0 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
84.35	18.01		0.10	5.00	0.10	
			Payment	43.05		
84.35	3.01		0.10	5.00	0.10	
			Delivery	4.04		
35.35	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
35.35	2.01		0.10	20.00	0.10	
			Order Status	4.04		
70.35	2.01		0.10	5.00	0.10	

7.5						
7.5 tt						
Key	RT	RT	Menu	Txn	Think	
Time	Delay	Fence	Delay	Weight	Time	
			New Order	44.83		
90.38	18.01		0.10	5.00	0.10	
			Payment	43.05		
90.38	3.01		0.10	5.00	0.10	
			Delivery	4.04		
37.88	2.01		0.10	5.00	0.10	
			Stock Level	4.04		
37.88	2.01		0.10	20.00	0.10	

Key	RT	RT	Menu	Txn	Think
75.38	2.01		Order Status	4.04	
			0.10	5.00	0.10
			8.0		
			8.0 tt		
Time	Delay	Fence	Delay	Weight	Time
96.40	18.01		New Order	44.83	
			0.10	5.00	0.10
96.40	3.01		Payment	43.05	
			0.10	5.00	0.10
40.40	2.01		Delivery	4.04	
			0.10	5.00	0.10
40.40	2.01		Stock Level	4.04	
			0.10	20.00	0.10
80.40	2.01		Order Status	4.04	
			0.10	5.00	0.10
			8.5		
			8.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
102.43	18.01		New Order	44.83	
			0.10	5.00	0.10
192.43	3.01		Payment	43.05	
			0.10	5.00	0.10
42.92	2.01		Delivery	4.04	
			0.10	5.00	0.10
42.92	2.01		Stock Level	4.04	
			0.10	20.00	0.10
85.42	2.01		Order Status	4.04	
			0.10	5.00	0.10
			9.0		
			9.0 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
108.45	18.01		New Order	44.83	
			0.10	5.00	0.10
108.45	3.01		Payment	43.05	
			0.10	5.00	0.10
45.45	2.01		Delivery	4.04	
			0.10	5.00	0.10
45.45	2.01		Stock Level	4.04	
			0.10	20.00	0.10
90.45	2.01		Order Status	4.04	
			0.10	5.00	0.10
			9.5		
			9.5 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
114.47	18.01		New Order	44.83	
			0.10	5.00	0.10

Key	RT	RT	Menu	Txn	Think
114.47	3.01		Payment	43.05	
			0.10	5.00	0.10
47.98	2.01		Delivery	4.04	
			0.10	5.00	0.10
47.98	2.01		Stock Level	4.04	
			0.10	20.00	0.10
95.47	2.01		Order Status	4.04	
			0.10	5.00	0.10
			10		
			10 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
120.50	18.01		New Order	44.83	
			0.10	5.00	0.10
120.50	3.01		Payment	43.05	
			0.10	5.00	0.10
50.50	2.01		Delivery	4.04	
			0.10	5.00	0.10
50.50	2.01		Stock Level	4.04	
			0.10	20.00	0.10
100.50	2.01		Order Status	4.04	
			0.10	5.00	0.10
			1.02 better		
			1.02 more aggressive		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.05	18.01		New Order	44.92	
			0.10	5.00	0.10
12.05	3.01		Payment	43.01	
			0.10	5.00	0.10
5.05	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.05	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.05	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.01 better		
			1.01 more aggressive		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.17	18.01		New Order	44.92	
			0.10	5.00	0.10
12.17	3.01		Payment	43.01	
			0.10	5.00	0.10
5.10	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.10	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.15	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.001 better		
			1.001 more aggressive		

Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.06	18.01		New Order	44.92	
			0.10	5.00	0.10
12.06	3.01		Payment	43.01	
			0.10	5.00	0.10
5.06	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.06	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.06	2.01		Order Status	4.02	
			0.10	5.00	0.10
			FullSpeed		
			1.000 tt		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.05	18.01		New Order	44.91	
			0.10	5.00	0.10
12.05	3.01		Payment	43.03	
			0.10	5.00	0.10
5.05	2.01		Delivery	4.02	
			0.10	5.00	0.10
5.05	2.01		Stock Level	4.02	
			0.10	20.00	0.10
10.05	2.01		Order Status	4.02	
			0.10	5.00	0.10
			1.003 best		
			1.003 best		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.09	18.01		New Order	44.90	
			0.10	5.00	0.10
12.09	3.01		Payment	43.05	
			0.10	5.00	0.10
5.07	2.01		Delivery	4.01	
			0.10	5.00	0.10
5.07	2.01		Stock Level	4.03	
			0.10	20.00	0.10
10.08	2.01		Order Status	4.01	
			0.10	5.00	0.10
			ExtraKick		
			FullSpeedKick		
Key	RT	RT	Menu	Txn	Think
				Weight	Time
Time	Delay	Fence	Delay	Weight	Time
12.03	18.01		New Order	44.93	
			0.10	5.00	0.10
12.03	3.01		Payment	43.01	
			0.10	5.00	0.10
5.03	2.01		Delivery	4.02	
			0.10	5.00	0.10

5.03	2.01		Stock Level	4.02	
		0.10	20.00	0.10	
10.03	2.01		Order Status	4.02	
		0.10	5.00	0.10	

ovd_11

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
10.85	18.00		New Order	44.92	
		0.10	5.00	0.10	
10.85	3.00		Payment	43.01	
		0.10	5.00	0.10	
4.55	2.00		Delivery	4.02	
		0.10	5.00	0.10	
4.55	2.00		Stock Level	4.03	
		0.10	20.00	0.10	
9.05	2.00		Order Status	4.02	
		0.10	5.00	0.10	

ovd_10

Key	RT	RT	Menu	Txn	Think
Time	Delay	Fence	Delay	Weight	Time
10.12	18.00		New Order	44.92	
		0.10	5.00	0.10	
10.12	3.00		Payment	43.01	
		0.10	5.00	0.10	
4.24	2.00		Delivery	4.02	
		0.10	5.00	0.10	
4.24	2.00		Stock Level	4.03	
		0.10	20.00	0.10	
8.44	2.00		Order Status	4.02	
		0.10	5.00	0.10	

HP Specific Drivers

The following Microsoft Windows 2008 Server x64 device drivers were replaced with HP-specific device drivers:
 The Microsoft HP Smart Array SAS Controller Controller default device driver (hpciss.SYS) was replaced with the HP Smart Array SAS Controller Non-miniport Performance Drivers for Microsoft Windows 2003/2008 Server x64 (hpqcissb.sys and hpqcissd.sys).

Appendix D: 60-Day Space

TPC-C 60 Day Space Requirements

Warehouses	54,000				TpmC	661,475
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	54,000	5,760	128	294		6,182
District	540,000	60,000	216	3,011		63,227
Customer	1,620,000,000	1,178,181,824	73,507,544	62,584,468		1,314,273,836
History	1,620,000,000	94,598,544	353,304		20,480,108	94,951,848
New_order	486,000,000	8,659,248	19,656	433,945		9,112,849
Orders	1,620,000,000	52,897,960	118,400		29,169,768	53,016,360
Order_line	16,199,947,865	1,062,291,664	2,501,712		374,072,604	1,064,793,376
Item	100,000	9,416	128	477		10,021
Stock	5,400,000,000	1,728,000,000	3,641,352	86,582,068		1,818,223,420
Total		4,124,704,416	80,142,440	149,604,264	423,722,480	4,354,451,120
		MB				
Dynamic Space	1,181,434	Sum of Data for Order, Orderline and History				
Static Space	3,070,960	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	231,552	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	16,964.096	MB				
60 Day Space GB	16,566.50	GB				
Log Size	2,074,069.00	MB				
KB Per New Order	6.50	KB				
8 hr log MB	2,016,897	MB				
8 hr log GB	1,969.63	GB				
Space Usage	GB Needed	Disks Measured	GB Priced	Disk Size	Formatted Size	
60 Day Space DB	16,566	1285	43,433.00	36GB	33.80	
		15	1,026.00	72GB	68.40	
			0.00			
Total DB			44,459.00			
8-hr log + mirror	3,939	32	4,374.40	146GB	136.70	
OS, Swap	3	2	67.60			
Total Storage	20,508.75	GB	48,901.00	GB		

MSSQL_stk_fg	MSSQL_cust_fg	MSSQL_ol_fg	MSSQL_misc_fg
			6,182
			63,227
	1,314,273,836		115,431,956
			9,112,849
		1,438,865,980	82,186,128
			10,021
1,818,223,420			
1,818,223,420	1,314,273,836	1,438,865,980	206,810,364
files=	13	13	13
size=	21,702,400	13,638,400	3,833,600
Total=	282,131,200	177,299,200	49,836,800
8K blocks	2,257,049,600	1,418,393,600	398,694,400
OK	OK	OK	OK

tpmC	661,475									
	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	Index Grow KB	Total Grow KB	KB/New-Order	8-Hr Growth KB	8-Hr Growth MB
History	94,598,544	353,304	103,204,864	702,360	8,606,320	349,056	8,955,376	0.0645	20,480,108.37	20,000.11
Order	52,897,960	118,400	65,536,416	235,064	12,638,456	116,664	12,755,120	0.0919	29,169,767.95	28,486.10
Order-Line	1,062,291,664	2,501,712	1,223,395,256	4,969,560	161,103,592	2,467,848	163,571,440	1.1782	374,072,603.67	365,305.28
										413,791.48
	sum(*) Before		sum(*) After		Num New-Order					
d_next_o_id	1,620,540,000		1,759,377,328		138,837,328					
	Before MB		After MB		Grow MB			KB/New-Order	8-Hr Growth MB	8-Hr Growth GB
Log	21,565.75		903,498.23		881,932.48			6.5047	2,016,897.19	1,969.63
								6,660.8400	bytes	
	2,153,930.00	1.001228	41.946499							
Database tpcc log used (%)										

Appendix E: *Third Party Letters*

7ft Pink Cat 6 Patch Cable, Molded - As low as 1.42 - Microsoft Internet Explorer

Address: http://www.deepsurplus.com/Network-Structured-Wiring/7-Foot-CAT-6-Patch-Cables/7ft-Pink-Cat-6-Patch-Cable-Molded-brfont-color="red">As-low-as-1-75-font

DeepSurplus
Surplus, Closeout & Overstocked Cabling Supplies
In business since 2002
Phone: 949-643-5004

Look what customers are saying about us:
eBay Rate
Google Product Search
Shopping.com
Yahoo! Shopping

HOME Network Cabling & Structured Wiring Home Theater (Audio/Video) Computer Cabling & Accessories Speaker Parts, Amplifier Building Electronic Components My Account Checkout Company Info

Network Cabling & Structured Wiring > Network Patch Cables > Ethernet CAT6 Network Patch Cables > Ethernet CAT6 Patch Cables; 7ft >

Register | Log In

Shopping Cart
Your Cart is Empty
View Cart

7ft Pink Cat 6 Patch Cable, Molded
As low as 1.42

Search

[Bulk Cable](#)
[Network Patch Cables](#)
[Ethernet CAT5e Network Patch Cables](#)
[Ethernet CAT5e Crossover Network Patch Cables](#)
[Ethernet CAT6 Network Patch Cables](#)
[Ethernet CAT6 Patch Cables; 1ft](#)
[Ethernet CAT6 Patch Cables; 2ft](#)
[Ethernet CAT6 Patch Cables; 3ft](#)
[Ethernet CAT6 Patch Cables; 5ft](#)
[Ethernet CAT6 Patch Cables; 7ft](#)
[7ft Pink Cat 6 Patch Cable, Molded](#)
As low as 1.42
[7ft Blue Cat 6 Patch Cable, Molded](#)
As low as 1.51
[7ft Black Cat 6 Patch Cable, Molded](#)
As low as 1.51
[7ft Green Cat 6 Patch Cable, Molded](#)
As low as 1.51
[7ft Gray Cat 6 Patch Cable, Molded](#)
As low as 1.51
[7ft Purple Cat 6 Patch Cable, Molded](#)

Add to cart to estimate shipping

Meets or exceeds the ANSI/TIA/EIA-568-B.2-1 standard for CAT 6 CMR, communication riser cable, and certified by UL, Underwriters Laboratories. Our CAT 6 patch cables come with a molded strain relief to protect the cable from tugs and pulls, special CAT 6 rated gold plated RJ45 connectors on each end and boots to protect the tab of the RJ45 connector from being snagged. Packaged individually in labeled bags.

P/N: CB242-7PK [Tell a Friend](#)
Condition: New
Mfg: Abergetty
P/N: CB242-7PK

Other great items you might enjoy:

Quantity	Price
1 - 499	\$1.80
500 - 749	\$1.52
750 - 999	\$1.47
1000 +	\$1.42

[7ft Yellow Cat 6 Patch Cable, Molded](#)
[5ft Blue Cat 6 Patch Cable, Molded](#)
[3ft Yellow Cat 6 Patch Cable, Molded](#)
[5ft Yellow Cat 6 Patch Cable, Molded](#)
[7ft Gray Cat 6 Patch Cable, Molded](#)

Discussions not available on http://www.deepsurplus.com/

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

January 27, 2010

Hewlett-Packard Company
David Adams
20555 SH 249
MS 150402
Houston, TX 77040

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 x64 Edition <i>Per Processor License</i> <i>Discount Schedule: Open Program - Level C</i> <i>Unit Price reflects a 6% discount from the retail unit price of \$24,999.</i>	\$23,432	2	\$46,864
P72-03168	Windows Server 2008 Enterprise Edition (x64) <i>Server License with 25 CALs</i> <i>Discount Schedule: Open Program - Level C</i> <i>Unit Price reflects a 42% discount from the retail unit price of \$3,999.</i>	\$2,310	1	\$2,310
P73-01972	Windows Server 2003 R2 Standard Edition <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 28% discount from the retail unit price of \$999.</i>	\$719	16	\$11,504
127-00012	Visual Studio Standard 2005 <i>Full License</i> <i>No Discount Applied</i>	\$250	1	\$250
N/A	Microsoft Problem Resolution Services <i>Professional Support</i> <i>(1 Incident)</i>	\$245	1	\$245

A list of Microsoft's resellers can be found

<http://www.microsoft.com/products/info/render.aspx?view=22&type=how>

All products listed above are currently orderable and available.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

This quote is valid for the next 90 days.

Reference ID: PCdaad1001270000005659.



November 3, 2009

David Adams
Hewlett-Packard / Houston
M/S M0704-402
20555 State Highway 249
Houston, TX 77070

Re: Quotation Number J9228

Dear David:

Thank you for your inquiry and the opportunity to supply this quotation. Product unit prices are listed below.

Product

600-2705 Seven-Slot PCI Express Expansion System

Includes PCIe-414 Seven-Slot PCI Express Expansion Backplane, PCIe-409 Repeater-Based PCI Express Host Bus to Expansion Cable Adapter, Rack Mount Chassis with 450 Watt Power Supply and One or Three Meter Expansion Cable.

Ordering P/N	Qty 1- 99
600-2705-1-SH	\$ 1,524
600-2705-3-SH	1,547

Delivery: Qty 1 –Two to Four Weeks after Receipt of Purchase Order
Country of Origin: USA
Warranty: One year. Repair on return-to-factory basis.
Payment Terms: Net 30 Days; Major Credit Cards Accepted
FOB: New Haven, CT, USA.
If possible, please supply UPS or FEDEX account number.

This quotation is valid and will be honored for ninety (90) days. After ninety days this quotation must be reissued. Current inventory levels are subject to change. To place this order, please contact our Sales Department, at 203-786-5536 and reference the quotation number.

If I can be of further assistance regarding this quotation or Cyclone Microsystems' products, please contact me by telephone at (203) 786-5536, by fax (203) 787-2323, or by email at peter.zackin@cyclone.com.

Sincerely,

Peter Zackin
Vice President, Sales

Cyclone Microsystems, Inc., 370 James Street, New Haven, CT 06513 USA Phone: 203 786-5536

Appendix F: *Price Verification*

All hardware is currently available..

HP Direct: 800-203-6748

For price verification before order date: e-mail hp.pricing.desk@hp.com