

TPC Benchmark™ C
Full Disclosure Report
for
IBM® System x3850 M2
using
Microsoft® SQL Server 2005
Enterprise x64 Edition (SP2)
and
Microsoft Windows® Server 2003 R2
Enterprise x64 Edition (SP2)

TPC-C Version 5.10

Submitted for Review
September 15, 2008

IBM Corporation

First Edition – September 2008

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is the customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

This publication was produced in the United States. IBM may not offer the products, services, or features discussed in this document in other countries, and the information is subject to change without notice. Consult your local IBM representative for information on products and services available in your area.

© Copyright International Business Machines Corporation 2008. All rights reserved.

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

U.S. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Trademarks

IBM, the IBM logo, System x, System Storage and TotalStorage are trademarks or registered trademarks of International Business Machines Corporation.

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark, tpmC, and \$/tpmC trademark of Transaction Processing Performance Council; Intel and Xeon are trademarks or registered trademarks of Intel Corporation; Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation. Other company, product, or service names, which may be denoted by two asterisks (**), may be trademarks or service marks of others.

Notes

¹ GHz and MHz only measures microprocessor internal clock speed, not application performance. Many factors affect application performance.

² When referring to hard disk capacity, GB, or gigabyte, means one thousand million bytes. Total user-accessible capacity may be less.

Abstract

IBM Corporation conducted the TPC Benchmark™ C on the IBM® System x3850 M2 configured as a client/server system with attached IBM System Storage™ DS4800. This report documents the full disclosure information required by the TPC Benchmark C Standard Specification, Revision 5.10, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

The software used on the x3850 M2 system includes Microsoft® Windows® Server 2003 Enterprise x64 Edition (SP2) and Microsoft SQL Server 2005 Enterprise x64 Edition (SP2).

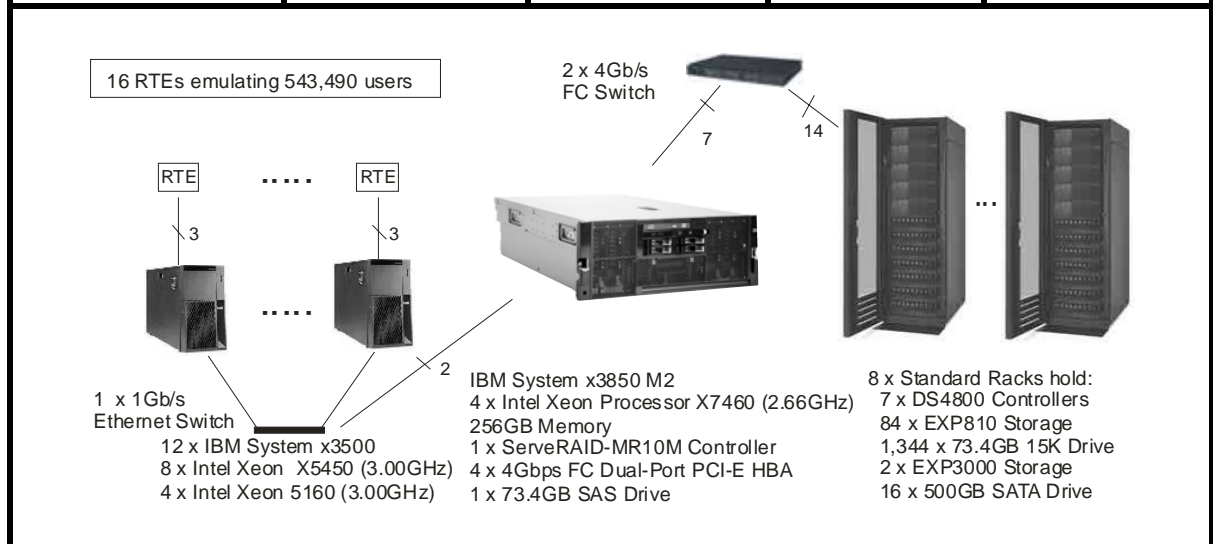
Two standard metrics, transactions per minute-C (tpmC) and price per tpmC (\$/tpmC), are reported as required by the TPC Benchmark C Standard Specification.

The benchmark results are summarized in the following table:

Hardware	Software	Total System Cost	tpmC	\$/tpmC	Total Solution Availability Date
IBM System x3850 M2	Microsoft SQL Server 2005 Enterprise x64 Edition (SP2) Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2)	\$1,763,438 USD	684,508	\$2.58 USD	October 31, 2008

The results of the benchmark and test methodology used were audited by Francois Raab of InfoSizing, Inc. The auditor's attestation letter is contained in Section 9 of this report.

IBM Corporation	IBM® System x3850 M2 Microsoft® SQL Server 2005 (SP2)		TPC-C Rev. 5.10
			Report Date 9/15/08
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
\$1,763,438 USD	684,508 tpmC	\$2.58 USD /tpmC	October 31, 2008
Database Server Processors/Cores/Threads	Database Manager	Operating System	Other Software
4/24/24 Intel® Xeon® Processor X7460 2.66 GHz	Microsoft SQL Server 2005 Enterprise x64 Edition (SP2)	Microsoft Windows® Server 2003 R2 Enterprise x64 Edition (SP2)	Microsoft Visual Studio Standard 2005 Microsoft COM+
			543,490



System Component	Qty	Server	Qty	Each of 12 Clients
Processors/Cores/Threads	4/24/24	Intel Xeon Processor X7460 (2.66GHz)	1/4/4	Quad-Core Intel Xeon X5450 (3.00GHz)
Cache	4	9MB L2, 16MB L3	1	2x6MB L2 Cache
Memory	32	8GB ECC RDIMM	4	512MB PC2-5300 FBD
Disk Controllers	7 4 1	IBM System Storage DS4800 IBM DS4000 4Gbps FC PCI-E ServeRAID-MR10M	1	Integrated SAS Interface
Disk Drives	1,344 16 1	73.4GB 15K 4Gbps FC 500GB H/S SATA II 73GB 15K SFF H/S SAS	1	146GB 15K SAS
Total Storage		99,261.89 GB		

Description	Part Number	Brand	Price Source	Unit Price	Quantity	Extended Price	3-Yr. Maint. Price
Server Hardware							
IBM System x3850 M2 (2 x Intel Xeon Processor X7460 with 2.66GHz/16MB L3 Cache, 4 Memory Cards, 8 x 1GB DIMM)	7233-6RU	IBM	1a *	19,539	1	19,539	
Intel Xeon Processor X7460 (2.66GHz/16MB L3 Cache)	44E4473	IBM	1a *	4,469	2	8,938	
16GB (2x8GB) 667MHz PC2-5300 ECC DDR2 SDRAM DIMM	43V7356	IBM	1	6,999	16	111,984	
ServeRAID-MR10M SAS/SATA Controller	43W4339	IBM	1	899	1	899	
73.4GB 15K Hot Swap SAS	43W7523	IBM	1	329	1	329	
IBM T115 15-inch TFT Display	494215U	IBM	1a	209	1	209	
IBM Preferred Pro USB Keyboard	40K9584	IBM	1	29	1	29	
IBM 3-Button Optical Mouse - Black - USB	40K9201	IBM	1	19	1	19	
NetXtreme II 1000 Express Ethernet Adapter	39Y6066	IBM	1	169	2	338	
ServicePac for 3-Year 24x7x4 Support (x3850 M2)	10N3059	IBM	1	1,695	1		1,695
ServicePac for 3-Year 24x7x4 Support (Display)	10N3110	IBM	1	90	1		90
Subtotal						142,284	1,785
Server Storage							
IBM 4Gb FC Dual-Port PCI-E HBA for IBM System x	39R6527	IBM	1	1,899	4	7,596	
IBM System Storage DS4800 Midrange Disk Subsystem	1815-82A	IBM	1	53,995	7	377,965	
4 Gbps SW SFP Transceiver 4 Pack	22R4897	IBM	1	550	33	18,150	
IBM 1m LC-LC Fibre Channel Cable	39M5696	IBM	1	79	168	13,272	
IBM 5m LC-LC Fibre Channel Cable	39M5697	IBM	1	129	21	2,709	
IBM System Storage DS4000 EXP810 Storage Exp. Unit	1812-81A	IBM	1	6,000	84	504,000	
16 PAK 73GB 15K 4Gbps FC E-DDM Hot-Swap HDD	1812-5433	IBM	1-S	14,816	84	1,244,544	
IBM TotalStorage SAN16B-2	200516B	IBM	1	4,120	2	8,240	
B16 4-Port Activation	22R4901	IBM	1	1,570	2	3,140	
IBM System Storage EXP3000	172701X	IBM	1	3,199	2	6,398	
500GB 3.5" Dual Port Hot-Swap SATA II	39M4558	IBM	1	399	16	6,384	
IBM EXP3000 1m Cable	39R6529	IBM	1	119	2	238	
IBM UPS 750TLV	2130R1X	IBM	1	299	1	299	
IBM S2 42U Standard Rack	93074RX	IBM	1	1,489	8	11,912	
ServicePac for 3-Year 24x7x4 Support (EXP810)	10N3651	IBM	1	960	84		80,640
ServicePac for 3-Year 24x7x4 Support (DS4800)	41C5953	IBM	1	3,200	7		22,400
ServicePac for 3-Year 24x7x4 Support (SAN16B-2)	41E9144	IBM	1	1,120	2		2,240
ServicePac for 3-Year 24x7x4 Support (EXP3000)	41L2768	IBM	1	760	2		1,520
ServicePac for 3-Year 24x7x4 Support (Rack)	41L2760	IBM	1	300	8		2,400
Subtotal						2,204,847	109,200
Server Software							
Microsoft SQL Server 2005 Enterprise x64 Edition	810-03134	Microsoft	2	24,999	4	99,996	
Microsoft Windows Server 2003 R2 Enterprise x64 Edition	P72-01684	Microsoft	2	3,999	1	3,999	
Microsoft Problem Resolution Services		Microsoft	2a	245	1		245
Subtotal						103,995	245
Client Hardware							
x3500 with 3.0GHz/12MB Xeon X5450, 1GB (2x512) Memory	7977R2U	IBM	1-S	3,459	12	41,508	
1GB (2x512MB) PC2-5300 CL5 ECC DDR2 FBD 667MHz	39M5782	IBM	1	129	12	1,548	
146GB 15K Hot Swap SAS	40K1044	IBM	1	359	12	4,308	
PRO/1000 PT Dual-Port Server Adapter	39Y6126	IBM	1	229	24	5,496	
ServicePac for 3-Year 24x7x4 Support (x3500)	21P2084	IBM	1	689	12		8,268
Subtotal						52,860	8,268
Client Software							
Microsoft Windows Server 2003 R2 Standard Edition	P73-01972	Microsoft	2	999	12	11,988	
Microsoft Visual Studio Standard 2005	127-00012	Microsoft	2a	250	1	250	
Subtotal						12,238	0
Network Components							
NetGear 24-Port Ethernet Switch (2 spares)	JGS524		3	225	3	675	
Ethernet Cable (2 spares)	A3L791-10-BLK		3	5	15	75	
Subtotal						750	0
Total						2,516,974	119,498
IBM Large Purchase Discount (See Note 1.)	34.60%		1			861,741	
Microsoft Open Program Discount Schedule (See Note 2.)	9.74%		2			11,293	

Pricing: 1 - IBM - 1-888-SHOP-IBM, ext. 5821; 2 - Microsoft; 3- newegg.com Note 1: Discount based on IBM Direct guidance applies to all line items where Pricing=1. Pricing is for this system or one of similar size. Note 2: Discount applies to all line items where Pricing=2. * These components are not immediately orderable. See the FDR for more information. S - One or more components of the measured configuration have been substituted in the Priced Configuration. See the FDR for details. Audited by Francois Raab, InfoSizing, Inc. (www.sizing.com)	Three-Year Cost of Ownership USD:	\$1,763,438
	tpmC:	684,508
	\$ USD/tpmC:	\$2.58

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

IBM Corporation	IBM® System x3850 M2 Microsoft® SQL Server 2005 (SP2)	TPC-C Rev. 5.10	
		Report Date 9/15/08	
Numerical Quantities Summary			
MQTh, Computed Maximum Qualified Throughput: 684,508 tpmC			
Response Times (in seconds)	90th Percentile	Average	Maximum
New Order	0.44	0.26	8.05
Payment	0.42	0.25	8.00
Delivery	0.14	0.13	6.95
Stock Level	0.80	0.55	8.49
Order Status	0.44	0.27	9.27
Delivery (Deferred)	0.13	0.10	4.94
Menu	0.13	0.13	6.95
Response Time Delay Added for Emulated Components: 0.1 Seconds			
Transaction Mix (in percent of total transactions)			Percent
New Order			44.95%
Payment			43.03%
Delivery			4.01%
Stock Level			4.01%
Order Status			4.01%
Keying/Think Times (in seconds)	Minimum	Average	Maximum
New Order	18.00 / 0.00	18.00 / 12.04	18.02 / 120.31
Payment	3.00 / 0.00	3.00 / 12.04	3.02 / 120.31
Delivery	2.00 / 0.00	2.00 / 5.03	2.02 / 50.31
Stock Level	2.00 / 0.00	2.00 / 5.04	2.02 / 50.31
Order Status	2.00 / 0.00	2.00 / 10.04	2.02 / 100.31
Test Duration			
Ramp-up Time			52 minutes 52 seconds
Measurement Interval			120 minutes
Number of Checkpoints			4
Checkpoint Interval			30 minutes
Number of transactions (all types) completed in measurement interval			190,066,699

Table of Contents

Abstract	3
Table of Contents	7
Preface	9
General Items	10
Benchmark Sponsor.....	10
Application Code Disclosure and Definition Statements	10
Parameter Settings	10
Configuration Diagrams	10
Clause 1 – Logical Database Design Related Items	12
Database Table Definitions.....	12
Database Physical Organization	12
Insert and Delete Definitions.....	12
Horizontal/Vertical Partitioning	12
Replication.....	12
Table Attributes	12
Clause 2 – Transactions and Terminal Profiles Related Items	13
Random Number Generation.....	13
Screen Layout	13
Terminal Verification	13
Intelligent Terminals.....	13
Transaction Profiles.....	13
Deferred Delivery Mechanism	14
Clause 3 – Transaction and System Properties Related Items	15
Atomicity Requirements.....	15
Consistency Requirements	15
Isolation Requirements	15
Durability Requirements.....	15
Clause 4 – Scaling and Database Population Related Items	17
Cardinality of Tables	17
Distribution of Tables and Logs	17
Database Model Implemented	18
Partitions/Replications Mapping	19
60-Day Space Requirement.....	19
Clause 5 – Performance Metrics and Response Time Items	20
Measured tpmC	20
Response Times.....	20
Keying/Think Times.....	20
Response Time Frequency Distribution Curves	20
Performance Curve for Response Time vs. Throughput.....	23
New Order Think Time Distribution	24
Steady State Methodology	25
Work Performed During Steady State	25
Measurement Interval	25
Transaction Mix.....	25
Percentage of Total Mix	25
Checkpoints	26
Clause 6 – SUT, Driver and Communication Related Items	27
Description of RTE.....	27
Emulated Components.....	27
Benchmarked and Targeted System Configuration Diagrams.....	27
Network Configuration.....	27
Network Bandwidth.....	27

Operator Intervention	27
Clause 7 – Pricing Related Items.....	28
Hardware and Software Components	28
Availability Date.....	28
Measured tpmC	28
Country-Specific Pricing	28
Usage Pricing.....	29
System Pricing	29
Clause 9 – Audit Related Items	30
Auditor.....	30
Availability of the Full Disclosure Report.....	30
Attestation Letter	30
Appendix A – Client Server Code	33
Web Client Source Code	33
Database Stored Procedures	155
Appendix B: Database Design Scripts	166
Database Build Source Code	166
Database Loader Source Code.....	178
Appendix C: Tunable Parameters	216
Database Configuration Parameters.....	216
Server Configuration Parameters.....	217
Storage Configuration Parameters	261
Client Configuration Parameters	269
RTE Input Parameters	301
Appendix D: 60-Day Space	304
Appendix E: Third-Party Price Quotes.....	305

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 Specification Version 5.10.

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

TPC Benchmark C is an On Line Transaction Processing (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 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.

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.

General Items

Benchmark Sponsor

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

This benchmark was sponsored by IBM Corporation.

Application Code Disclosure 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 and 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 that have been changed from the defaults found in actual products, including but not limited to:

- *Database tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*
- *Compilation and linkage options and run-time optimizations used to create/install applications, OS, and/or databases.*

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

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

Configuration Diagrams

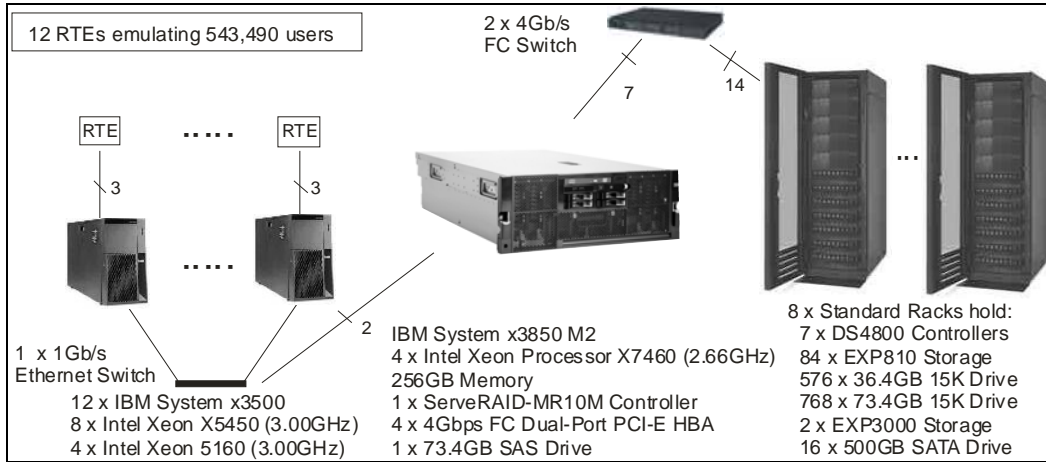
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The configuration diagram for the tested system is provided on the following page. See the Executive Summary for the priced configuration.

The Remote Terminal Emulator (RTE) used for these TPC Benchmark C tests is an IBM proprietary RTE. Under Version 5.10, the components of the configuration being emulated by the RTE are the workstations and the Ethernet hubs. Appendix C contains a listing of the RTE scripts and inputs used in the benchmark testing.

The tested configuration used 12 IBM System x3500 systems as the clients, which executed the terminal I/O and submitted transactions to COM+ servers, which are also running on the clients. These COM+ servers forwarded the transaction requests to the server, and returned the results to the RTE. Microsoft SQL Server 2005 Enterprise x64 Edition is the DBMS executing on the server. Eight of the x3500 clients were configured with one quad-core 3.00GHz Intel Xeon X5450 processor, and the other four x3500 clients were configured with one dual-core 3.00GHz Intel Xeon 5160 processor. All of the x3500 clients had 2GB of memory, one 146GB 15K SAS drive, six expansion slots, and two dual-port Gb Ethernet adapters.

Measured Configuration



The measured and priced configurations differed in two areas:

1) The priced configuration contained 12 clients, all of which were x3500 systems containing one quad-core 3.00GHz Intel Xeon X5450 processor. The measured configuration contained eight of the priced clients plus four x3500 systems containing one dual-core 3.00GHz Intel Xeon 5160 processor. Other than the CPU, all 12 measured clients were identical. Client processor substitution was based on the following:

	x3500 Quad-Core (Priced)	x3500 Dual-Core
Number of Clients	8	4
Processors Per Client	1	1
CPU Speed	3.00GHz	3.00GHz
Number of Cores	4	2
Bus Speed	1333MHz	1333MHz
Cache (L1-I, L1-D, L2)	4x32K, 4x32K, 2x6144k	2x32k, 2x32k, 2x4096k
Cache per Core	32k, 32k, 3072k	32k, 32k, 2048k

2) The priced configuration contained 1,344 73GB 4Gb 15K FC disk drives. The measured configuration contained 768 of the priced FC drives and 576 36GB 4Gb 15K FC disk drives. Disk substitution was based on the following:

	73GB (Priced)	36GB
Number Used	768	576
Model	ST373455FC	ST336854FC
Capacity	73.4GB	36.7GB
Interface	4Gb/s FC	4Gb/s FC
Track-to-Track Seek	0.2ms	0.2ms
Average Seek	3.5ms	3.5ms
Buffer Size	16MB	16MB
RPM	15000	15000
Media Density (peak)	890Kb/inch	628Kb/inch

For the priced configuration, see the Executive Summary.

Clause 1 – Logical Database Design Related Items

Database Table Definitions

Listings 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.

Database Physical Organization

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

Physical space was allocated to Microsoft SQL Server 2005 on the server disks as detailed in Table 4-2.

Insert and Delete Definitions

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 restriction in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

All insert and delete functions were fully operational during the running of the benchmark. The space required for an additional 5 percent of the initial table cardinality was allocated to Microsoft SQL Server 2005 and priced as static space.

Horizontal/Vertical Partitioning

While there are few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark (see Clause 1.6), any such partitioning must be disclosed.

Partitioning was not used in this benchmark.

Replication

Replication tables, if used, must be disclosed (see Clause 1.4.6).

Replication was not used in this benchmark.

Table Attributes

Additional and/or duplicated attributes in any table must be disclosed, along with a statement on the impact on performance (see Clause 1.4.7).

No additional attributes were used in this benchmark.

Clause 2 – Transactions and Terminal Profiles Related Items

Random Number Generation

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

The seeds and offsets for the random number generator were collected and verified to be different for each driver. The auditor selected samples of the generated numbers from the database. The samples were verified to have no discernible patterns.

Screen Layout

The actual layouts of the terminal input/out screens must be disclosed.

All screen layouts followed the TPC Benchmark C Standard Specification.

Terminal 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 must for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The auditor verified terminal features by direct experimentation. The benchmarked configuration uses Microsoft Internet Explorer 6.0 SP1 and HTML scripts as the terminal interface.

Intelligent Terminals

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

The terminals emulated in the priced configuration are desktop computer systems. All processing of the input/output screens was handled by the x3500 clients. The screen input/output was managed via HTML strings that comply with the HTML Version 2.0 specification. A listing of the code used to implement the intelligent terminals is provided in Appendix A. All data manipulation was handled by the x3850 M2 database server.

Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed.

The number of items per orders entered by New-Order transactions must be disclosed. The percentage of home and remote Payment transactions must be disclosed. The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed.

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed.

Table 2-1. Transaction Statistics

New Order	Values
Home warehouse order lines	99.00%
Remote warehouse order lines	1.00%
Rolled back transactions	1.00%
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	85.00%
Remote warehouse payment transactions	15.00%
Non-Primary Key Access	
Payment transactions using C_LAST	60.00%
Order-Status transactions using C_LAST	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.95%
Payment	43.03%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

Deferred Delivery Mechanism

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

The Delivery transaction was submitted to an ISAPI queue that is separate from the COM+ queue that the other transactions used. This queue is serviced by a variable amount of threads that are separate from the worker threads inside the web server. Web server threads are able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the queue are responsible for completing the deferred part of the transaction asynchronously.

The source code is listed in Appendix A.

Clause 3 – Transaction and System Properties Related Items

The results of the ACID test 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.

Atomicity Requirements

The system under test must guarantee that 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.

All ACID tests were conducted according to specification.

Completed Transactions

The following steps were performed to verify the Atomicity of completed transactions:

1. The balance was retrieved from the CUSTOMER table for a random Customer, District and Warehouse, giving BALANCE_1.
2. The Payment transaction was executed for the Customer, District and Warehouse used in step 1.
3. The balance was retrieved again for the Customer used in step 1 and step 2, giving BALANCE_2. It was verified that BALANCE_1 was greater than BALANCE_2 by AMT.

Aborted Transactions

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was changed to execute a rollback of the transaction instead of performing the commit.
2. Using the balance, BALANCE_2, from the CUSTOMER table retrieved for the completed transaction, the Payment transaction was executed for the Customer, District and Warehouse used in step 1. The transaction rolled back due to the change in the application code from step 1.
3. The balance was retrieved again for the Customer used for step 2, giving BALANCE_3. It was verified that BALANCE_2 was equal to BALANCE_3.

Consistency Requirements

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 batch file to issue queries to the database. The results of the queries demonstrated that the database was consistent for all four tests.

Isolation Requirements

Sufficient conditions must be enabled at either the system or the application level to ensure that the required isolation defined in Clause 3.4.1 is obtained.

Isolation tests one through nine were run using batch files to issue queries to the database. Each file included timestamps to demonstrate the concurrency of operations. The results of the queries were captured and placed in files. The auditor reviewed the results and verified that the isolation requirements had been met.

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

Case A was followed for isolation test seven.

Durability Requirements

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

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (this test includes failure of all or part of memory)*
- *Instantaneous interruption (system crash/system hang) in processing that requires system reboot to recover*
- *Failure of all or part of memory (loss of contents)*

Loss of Data Test

The FibreChannel disks attached to each DS4800 controller were configured as RAID-0 arrays, with each array containing a portion of each of the tables in the database.

The following steps were successfully performed to pass the loss of data durability test with database tables:

1. The contents of the database were backed up to several database dump devices during the initial database load.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
3. A test was started with 54,350 users submitting transactions.
4. The benchmark ran for 5 minutes after all users were connected to the server.
5. A Fibre Channel disk drive containing a portion of each of the tables in the tpc database was pulled out. This caused Microsoft SQL Server 2005 to report errors accessing that device since the data arrays are all RAID-0.
6. The run was aborted and Microsoft SQL Server 2005 was stopped.
7. The database log was saved.
8. A new disk drive was reinserted and was recovered.
9. The database was restored from the backup devices. Afterwards, the saved database transaction log was rolled forward.
10. Step 2 was repeated to obtain the current count of the total number of orders giving SUM2.
11. It was verified that SUM2 minus SUM1 (the actual number of new orders added to the database during the test) was equal to the number of new order transactions completed during the run minus any rollback transactions, as reported by the RTE logs.
12. Consistency Condition 3 was verified.

Loss of Log and System (Instantaneous Interruption and Loss of Memory) Tests

The SATA disks attached to the ServeRAID-MR10 controller were configured as one RAID-10 array. This array held the database log. Write-caching was disabled for the log controller.

The following steps were successfully performed to pass the loss of log drive and loss of system power durability tests:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID for all rows in the district table giving SUM1.
2. A test was started with 543,490 users submitting transactions.
3. The benchmark ran for 5 minutes after all users were connected to the server.
4. One disk from the log array was removed. Since the log array was RAID-10 mirrored, Microsoft SQL Server 2005 continued to process transactions without interruption.
5. The test continued to run for another 5 minutes.
6. The server under test was powered off, which removed power from the system and the memory.
7. The server was powered on again.
8. Microsoft SQL Server 2005 was allowed to recover.
9. Step 1 was repeated to obtain the current count of the total number of orders giving SUM2.
10. It was verified that SUM2 minus SUM1 (the actual number of new orders added to the database during the test) was greater than or equal to the number of new order transactions completed during the run minus any rollback transactions, as reported by the RTE logs.

Clause 4 – Scaling and Database Population Related Items

Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run (see Clause 4.2), must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted (see Clause 4.2.2), the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The database was built with 56,000 warehouses and the audited run used 54,349 warehouses. A total of 1,651 warehouses was deleted.

Table 4-1. Initial Cardinality of Tables

Table Name	Rows
Warehouse	56,000
District	560,000
Item	100,000
New Order	504,000,000
History	1,680,000,000
Orders	1,680,000,000
Customer	1,680,000,000
Order Line	16,799,949,701
Stock	5,600,000,000
Inactive Warehouses	1,651

Distribution of Tables and Logs

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

The log was configured using one ServeRAID-MR10M controller. This controller was connected to a RAID-10 disk array consisting of sixteen 500GB hot-swap SATA II disks housed in two EXP3000 drive enclosures.

The database tables were stored on 1,344 73.4GB 15K RPM 4Gb Fibre Channel disk drives. These disks were housed in EXP810 enclosures, which hold 16 drives each. A total of seven DS4800 storage systems were used to control these disks, each connected to 192 disks.

Each disk volume, as seen by the Windows operating system, was configured as a RAID-0 disk array with 96 physical disks, for a total of two RAID-0 arrays per DS4800. There was a total of 14 such disk volumes. Each disk volume was configured in Windows as two partitions, one for customer and stock and the other for misc.

Table 4-2 depicts the database configuration of the measured and priced systems to meet the 8-hour steady state requirement.

Table 4-2. Data Distribution for the Benchmarked Configuration

Disk #	Drives	Partition	Size	Use
0	1 – 73GB SAS	C:	68.36GB (NTFS)	Operating System
1	16 - 500GB EXP3000 Enclosures	E: G:	1953.61GB 488.28GB	Log Log
2	96 - 73GB EXP810 Enclosures	C:\mp\c1 C:\mp\m1	225.10GB 122.56GB	Customer and Stock Miscellaneous
3	96 - 73GB EXP810 Enclosures	C:\mp\c2 C:\mp\m2	225.10GB 122.56GB	Customer and Stock Miscellaneous
4	96 - 73GB EXP810 Enclosures	C:\mp\c3 C:\mp\m3	225.10GB 122.56GB	Customer and Stock Miscellaneous
5	96 - 73GB EXP810 Enclosures	C:\mp\c4 C:\mp\m4	225.10GB 122.56GB	Customer and Stock Miscellaneous
6	96 - 73GB EXP810 Enclosures	C:\mp\c5 C:\mp\m5	225.10GB 122.56GB	Customer and Stock Miscellaneous
7	96 - 73GB EXP810 Enclosures	C:\mp\c6 C:\mp\m6	225.10GB 122.56GB	Customer and Stock Miscellaneous
8	96 - 73GB EXP810 Enclosures	C:\mp\c7 C:\mp\m7	225.10GB 122.56GB	Customer and Stock Miscellaneous
9	96 - 73GB EXP810 Enclosures	C:\mp\c8 C:\mp\m8	225.10GB 122.56GB	Customer and Stock Miscellaneous
10	96 - 73GB EXP810 Enclosures	C:\mp\c9 C:\mp\m9	225.10GB 122.56GB	Customer and Stock Miscellaneous
11	96 - 73GB EXP810 Enclosures	C:\mp\c10 C:\mp\m10	225.10GB 122.56GB	Customer and Stock Miscellaneous
12	96 - 73GB EXP810 Enclosures	C:\mp\c11 C:\mp\m11	225.10GB 122.56GB	Customer and Stock Miscellaneous
13	96 - 73GB EXP810 Enclosures	C:\mp\c12 C:\mp\m12	225.10GB 122.56GB	Customer and Stock Miscellaneous
14	96 - 73GB EXP810 Enclosures	C:\mp\c13 C:\mp\m13	225.10GB 122.56GB	Customer and Stock Miscellaneous
15	96 - 73GB EXP810 Enclosures	C:\mp\c14 C:\mp\m14	225.10GB 122.56GB	Customer and Stock Miscellaneous

Database Model Implemented

A statement must be provided that describes:

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

Microsoft SQL Server 2005 Enterprise Edition is a relational database. The interface used was Microsoft SQL Server stored procedures accessed with Remote Procedure Calls embedded in C code using the Microsoft ODBC interface.

Partitions/Replications Mapping

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

Neither horizontal nor vertical partitioning was used.

60-Day Space Requirement

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 (see Clause 4.2.3).

See Appendix D for details about how the 60-day space requirements were calculated.

Clause 5 – Performance Metrics and Response Time Items

Measured tpmC

Measured tpmC must be reported.

Measured tpmC: 684,508 tpmC

Price per tpmC: \$2.58 USD per tpmC

Response Times

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

The TPC-C requirements for the average response time and the 90th percentile were met. Table 5-1 provides the response times for each of the transaction types and the menu for the measured system.

Table 5-1. Response Times in Seconds

Transaction Type	Average	Maximum	90 %-tile
New-Order	0.26	8.05	0.44
Payment	0.25	8.00	0.42
Delivery	0.13	6.95	0.14
Stock Level	0.55	8.49	0.80
Order Status	0.27	9.27	0.44
Delivery (Deferred)	0.10	4.94	0.13
Menu	0.13	6.95	0.13

Keying/Think Times

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

Table 5-2 lists the keying/think times for the measured system.

Table 5-2. Keying/Think Times in Seconds

Transaction Type	Minimum	Average	Maximum
New-Order	18.00 / 0.00	18.00 / 12.04	18.02 / 120.31
Payment	3.00 / 0.00	3.00 / 12.04	3.02 / 120.31
Delivery	2.00 / 0.00	2.00 / 5.03	2.02 / 50.31
Stock Level	2.00 / 0.00	2.00 / 5.04	2.02 / 50.31
Order Status	2.00 / 0.00	2.00 / 10.04	2.02 / 100.31

Response Time Frequency Distribution Curves

Response time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type.

Figure 5-1. New-Order Transaction - Response Time Frequency Distribution

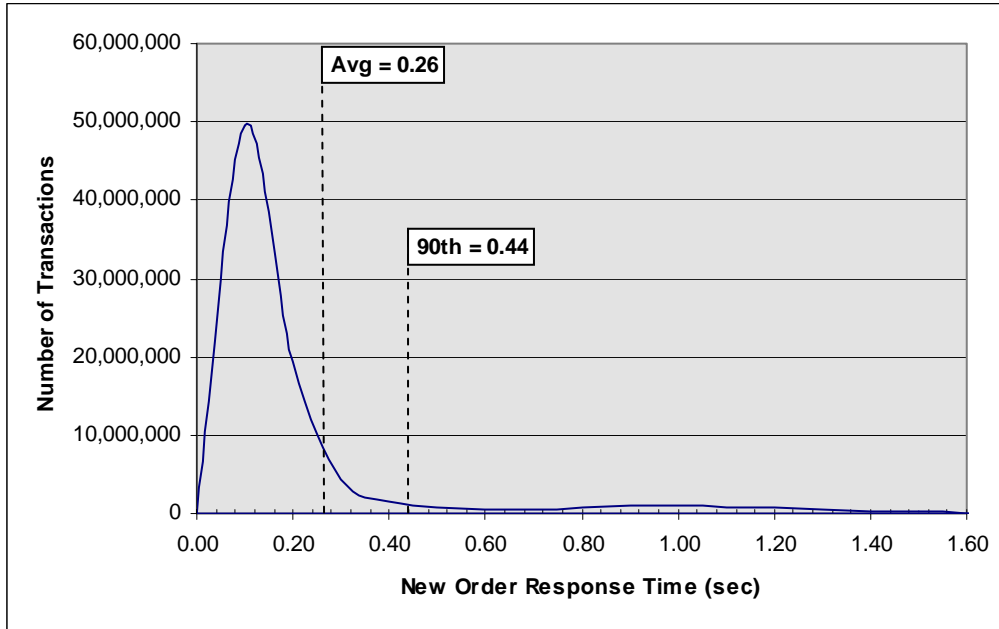


Figure 5-2. Payment Transaction - Response Time Frequency Distribution

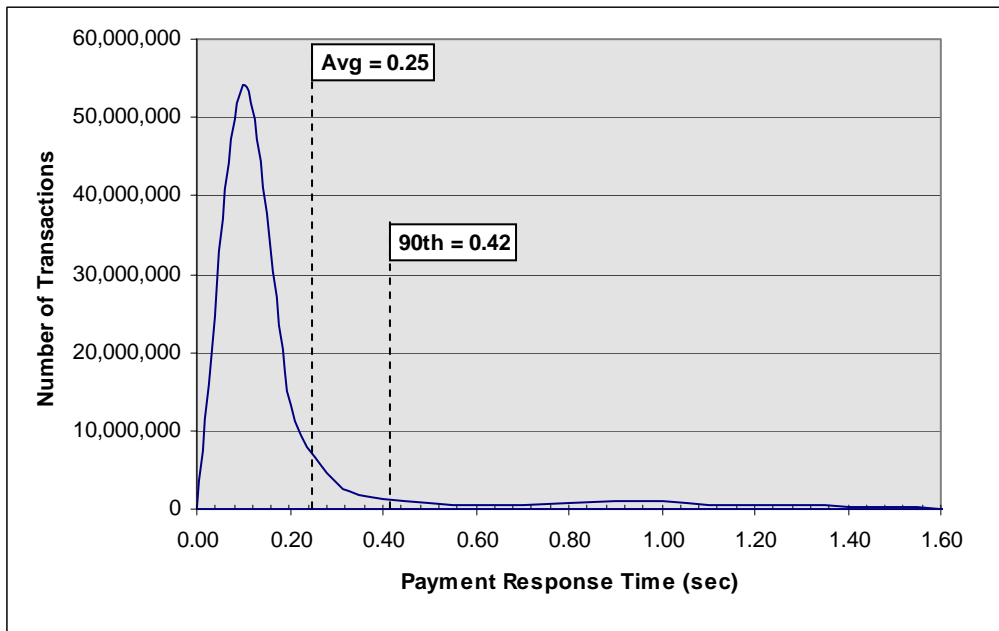


Figure 5-3. Order-Status Transaction - Response Time Frequency Distribution

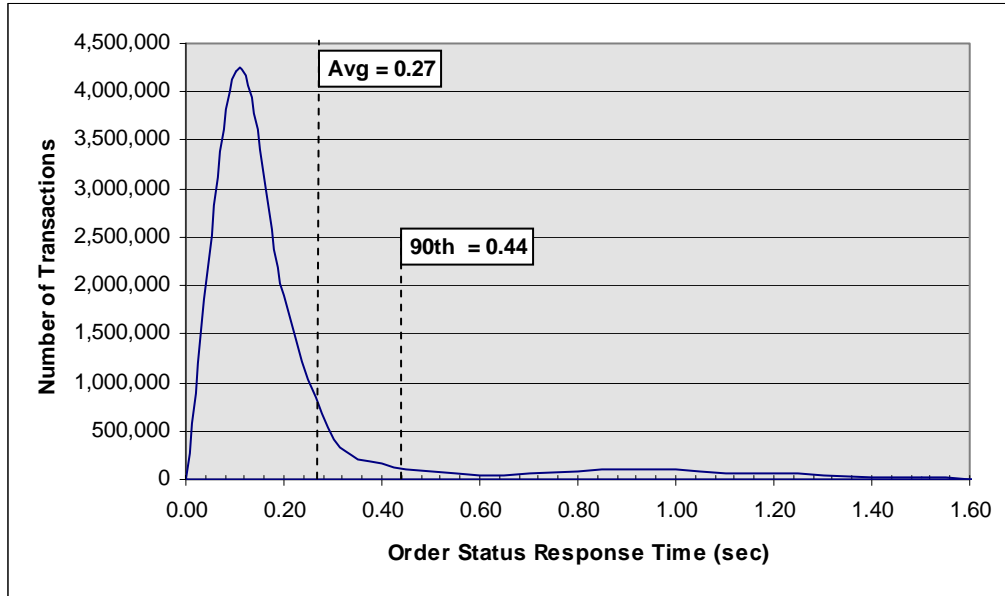


Figure 5-4. Delivery Transaction - Response Time Frequency Distribution

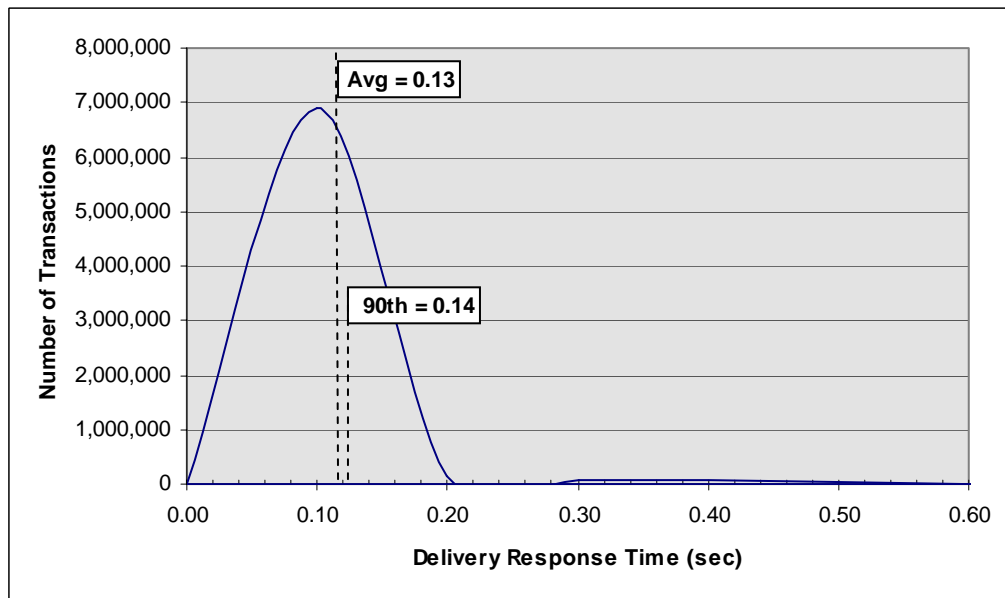
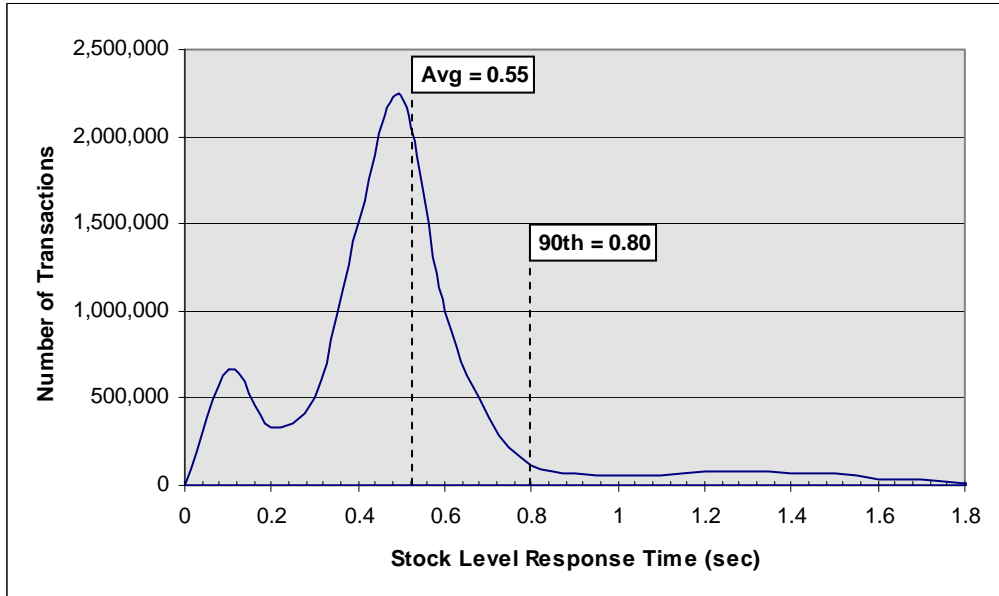


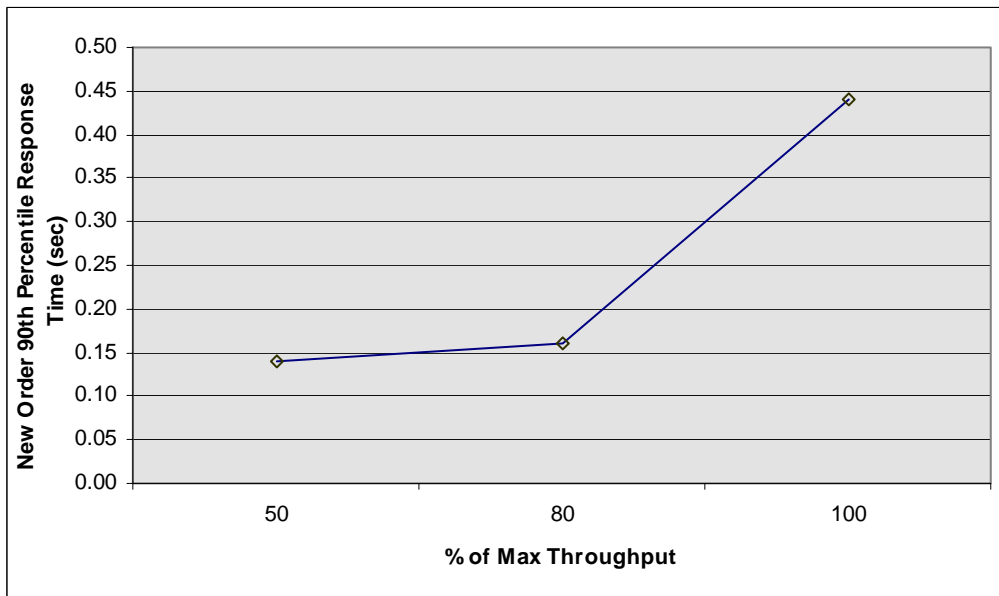
Figure 5-5. Stock-Level Transaction - Response Time Frequency Distribution



Performance Curve for Response Time vs. Throughput

The performance curve for response time vs. throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

Figure 5-6. New-Order Response Time vs. Throughput



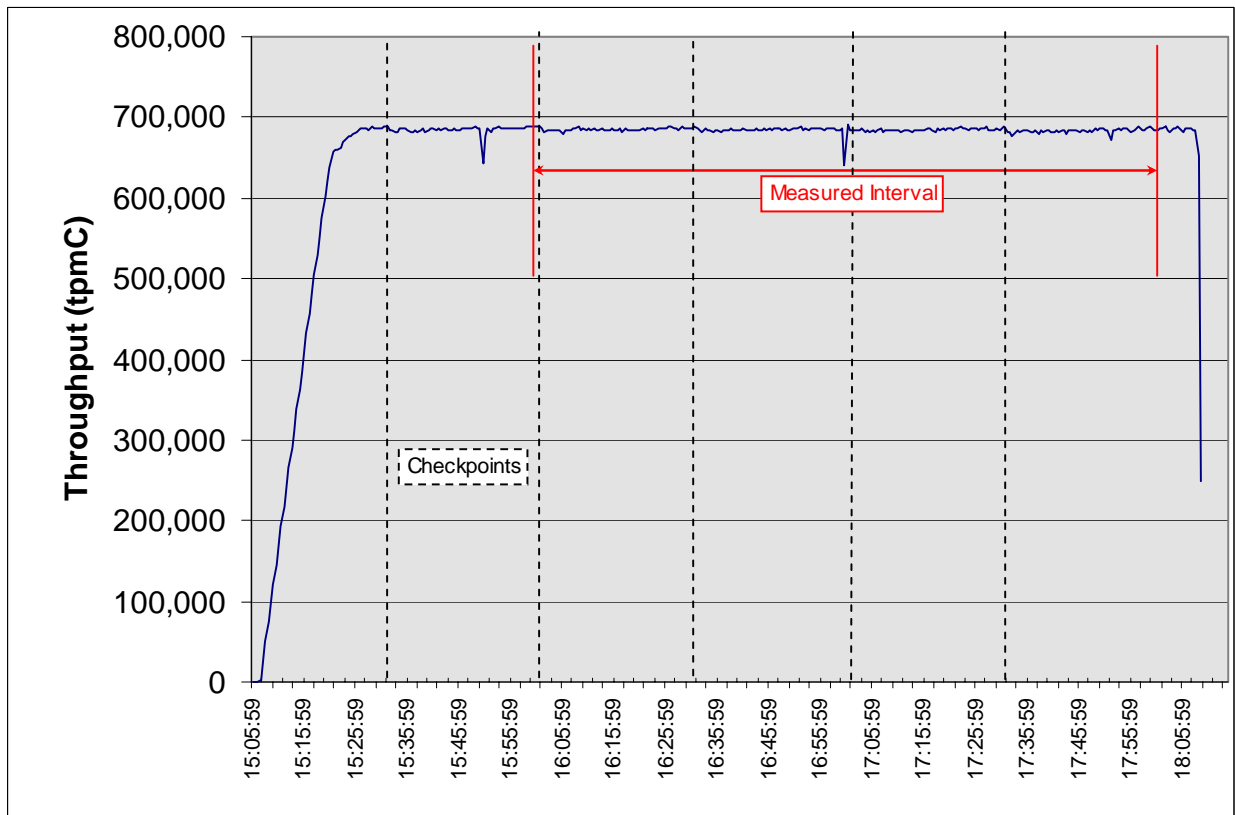
New Order Think Time Distribution

Figure 5-7. New-Order Think Time Distribution



A graph of throughput vs. elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

Figure 5-8. New-Order Throughput vs. Elapsed Time



Steady State Methodology

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval (see Clause 5.5) must be described.

Figure 5-8 shows that the system was in steady state at the beginning of the measurement interval.

Work Performed During Steady State

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

Transaction Flow

The RTE generated the required input data to choose a transaction from the menu. This data was time-stamped. The response for the requested transaction was verified and time-stamped in the RTE log files.

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the 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 time-stamps was the response time for that transaction and was logged in the RTE log. The RTE then waited the required think time interval before repeating the process starting at selecting another 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 a Gigabit Ethernet network using Microsoft SQL Server ODBC library and RPC calls.

Measurement Interval

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

The measurement interval was 120 minutes.

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. (8.1.6.13)

The RTE was given a weighted random distribution, which was not adjusted during the run. See Table 5-3.

Percentage of Total Mix

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

See Table 5-3.

Table 5-3. Transaction Statistics and Transaction Mix

New Order	Values
Home warehouse order lines	99.00%
Remote warehouse order lines	1.00%
Rolled back transactions	1.00%
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	85.00%
Remote warehouse payment transactions	15.00%
Non-Primary Key Access	
Payment transactions using C_LAST	60.00%
Order-Status transactions using C_LAST	60.00%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.95%
Payment	43.03%
Delivery	4.01%
Stock Level	4.01%
Order Status	4.01%

Checkpoints

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.

During a checkpoint, SQL Server flushes all dirty pages from its cache to disk. It places a record in the database transaction log indicating that the checkpoint has completed and that all transactions which were committed prior to the checkpoint have been written to disk. Checkpoints were performed during the ramp-up period and during each measured run interval. SQL Server was started with trace flag 3502, which caused it to log the occurrence of the checkpoints. This information was used to verify that the checkpoints occurred at the appropriate times during the test run.

The first measurement interval checkpoint started 17 seconds after the start of the measurement interval. The checkpoint interval was 30 minutes and each checkpoint lasted for 28 minutes 20 seconds. The checkpoints in the measured interval are shown in Table 5-4.

Table 5-4. Checkpoint Timing

Checkpoint	Start Time	Duration
1	16:01:25	28 minutes 20 seconds
2	16:31:20	28 minutes 20 seconds
3	17:01:15	28 minutes 20 seconds
4	17:31:10	28 minutes 20 seconds

Clause 6 – SUT, Driver and Communication Related Items

Description of RTE

The RTE input parameters, code fragments, functions, etc., used to generate each transaction input field must be disclosed.

The RTE used is IBM-developed proprietary software. The RTE input is listed in Appendix C.

Emulated Components

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

No components were emulated.

Benchmarked and Targeted System Configuration Diagrams

A complete functional diagram of both the benchmarked configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all software and hardware functionality being performed on the Driver System, and its interface to the SUT must be disclosed (see Clause 6.6.3.6).

The driver RTE generated the transaction input data and transmitted it to the client in HTML format. The driver RTE received the output from the System Under Test, time-stamped it, and forwarded it to the Master RTE for post-test processing. No other functionality was included on the driver RTE.

Detailed diagrams of the benchmarked and priced configurations are provided in the section called “General Items” at the beginning of this document.

Network Configuration

The network configurations of both the tested services and the proposed (target) services which 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 (see Clause 6.6.4).

See the measured and priced configuration diagrams for details about the network configuration.

Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed.

The Ethernet used in the LAN connecting the clients and driver RTEs complies with the IEEE.802.3 standard. The Ethernet LAN had a bandwidth of 1Gbps. The LAN that connected the clients to the server complies with the IEEE.802.3 standard. The Ethernet LAN had a bandwidth of 1Gbps.

Operator Intervention

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

The configuration did not require any operator intervention to sustain the reported throughput.

Clause 7 – Pricing Related Items

Hardware and Software Components

A detailed list of the hardware and software used in the priced system must be reported. Each separately orderable item must have a vendor part number, description and release/revision level, and either general availability status or committed delivery date. 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(s) and effective date(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.

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix E at the end of this document.

Availability Date

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

The total solution as priced will be generally available October 31, 2008. The dates for ordering and availability are detailed in Table 7-1 for those components not immediately orderable.

Table 7-1. Ordering and Pricing Information

Description	Part Number	Order Date	Availability Date	Order Method	Price Verification
System x3850 M2 with Intel Xeon Processor X7460	7233-6RU	9-16-08	10-31-08	See Note 1	See Note 2
Intel Xeon Processor X7460	44E4473	9-16-08	10-31-08	See Note 1	See Note 2

Note 1: IBM – 1-888-746-7426, ext. 5821

Note 2: These components are not immediately orderable. For price verification before order date, call 1-888-746-7426, ext. 5821.

Measured tpmC

A statement of the measured tpmC, as well as the respective calculations for the 3-year pricing, price/performance (price/tpmC) and the availability date must be included.

- Maximum Qualified Throughput: 684,508 tpmC
- Price per tpmC: \$2.58 USD per tpmC
- Three-year cost of ownership: \$1,763,438 USD

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.

The configuration is 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:

- 1 Microsoft Windows Server 2003 R2 Enterprise x64 Edition
- 12 Microsoft Windows Server 2003 R2 Standard Edition
- 4 Microsoft SQL Server 2005 Enterprise x64 Edition (SP2) (based on per-processor price)
- 3-year support for hardware components (except for components for which a minimum of 2 or 10 percent spares are provided)

System Pricing

System pricing should include subtotals for the following components: Server Hardware, Server Software, Client Hardware, Client Software, and Network Components used for terminal connection (see Clause 7.2.2.3). System pricing must include line item indication where non-sponsoring companies' brands are used. System pricing must also include line item indication of third-party pricing.

A detailed list of all hardware and software, including the 3-year price, is provided in the Executive Summary at the front of this report. All third-party quotations are included in Appendix E at the end of this document.

Clause 9 – Audit Related Items

Auditor

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-C benchmark was audited by Francois Raab of InfoSizing, Inc. The auditor's attestation letter is provided in this section.

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 BenchmarkTMC," the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same.

The TPC Benchmark C Full Disclosure Report can be obtained from www.tpc.org.

Attestation Letter

The auditor's Attestation Letter is on the next two pages.

Celia Schreiber
 Manager, System x Performance Analysis and Benchmarking
 3039 Cornwallis Road
 Research Triangle Park, NC 27709

September 5, 2008

I verified the TPC Benchmark™ C performance of the following Client Server configuration:

Platform: IBM System x3850 M2
 Operating system: Microsoft Windows Server 2003 R2 Enterprise x64 Edition (SP2)
 Database Manager: Microsoft SQL Server 2005 Enterprise x64 Edition (SP2)
 Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	New Order 90% Response Time	tpmC
Server: IBM System x3850 M2				
4 x Intel Xeon X7460 (2.66GHz)	256 GB (4 x 16MB L3)	1344 x FC 73.4 GB 15K rpm 16 x 500 GB SATA II	0.44 Seconds	684,508.68
12 Clients: IBM System x3500 (Specification for each)				
1 x Intel Xeon X5450 QC (3.00 GHz)	2 GB (2 x 6 MB L2)	1 x 146 GB 15K rpm SAS	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 120 minutes
- Four checkpoints were taken during the measurement interval
- The 60 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

Disk Substitution

The tested configuration included (768) priced 73.4 GB disk drives and (576) non-priced 36.4 GB disk drives. The priced configuration replaces the (576) 36.4 GB disk drives with (576) 73.4 GB disk drives. Based on the specifications of these disks, it is my opinion that this substitution has no significant effect on performance.

Client Substitution

The tested configuration included (8) priced clients model IBM System x3500 (Intel Xeon X5450) and (4) non-priced clients model IBM System x3500 (Intel Xeon 5160). The priced configuration replaces the (4) IBM System x3500 (Intel Xeon 5160) clients with (4) IBM System x3500 (Intel Xeon X5450) clients. Based on data analysis done for each type of client, it is my opinion that this substitution has no significant effect on performance.

Respectfully Yours,

A handwritten signature in black ink, appearing to read 'François Raab', written in a cursive style.

François Raab, President

Appendix A – Client Server Code

Web Client Source Code

tpcc_oledb.cpp

```
/*      FILE:          TPCC_OLEDB.CPP
 *
 *      Kit Ver. 4.42.000
 *
 *      2004
 *
 *      Written by Sergey
 *      Vasilevskiy
 *      All Rights Reserved
 *
 *      PURPOSE:       Implements OLEDB calls for
 *      TPC-C txns.
 *      Contact:   Charles Levine (clevine@microsoft.com)
 *
 */

#include <windows.h>
#include <stdio.h>
#include <assert.h>
#include <stddef.h>

#define DBINITCONSTANTS
#include <oledb.h>
// #include <sqloledb.h> // Use MDAC
#include <sqlncli.h> // Use SNAC
#include <oledberr.h>

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

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

#include "..\..\common\src\error.h"
#include "..\..\common\src\trans.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_oledb.h"
```

```
#ifndef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

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

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

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

// this needs to be the same as the max length of
machine/database/user/password in Benchcraft (engstut.h)
const static int iMaxNameLen = 32;

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

            DisableThreadLibraryCalls(hModule);

            break;

        case DLL_PROCESS_DETACH:

            break;

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

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
 *
 */

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

    static SERRORMSG errorMsgs[] =
    {
        { ERR_WRONG_SP_VERSION,
          "Wrong version of stored procs on
          database server"
        },
        { ERR_INVALID_CUST,
          "Invalid Customer id,name."
        }
    }
}
```

```
},
    { ERR_NO_SUCH_ORDER,
      "No orders found for customer."
    },
    { ERR_RETRIED_TRANS,
      "Retries before transaction succeeded."
    },
    { 0,
      ""
    }
};

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

for(i=0; errorMsgs[i].szMsg[0]; i++)
{
    if ( m_errno == errorMsgs[i].iError )
        break;
}
if ( !errorMsgs[i].szMsg[0] )
    return szNotFound;
else
    return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec( dllexport ) CTPCC_OLEDB* CTPCC_OLEDB_new(
LPCSTR szServer, // name of SQL
server LPCSTR szUser, // user
name for login LPCSTR szPassword, // password for
login LPCSTR szHost, // not
used LPCSTR szDatabase, // name of database
to use LPCWSTR szSPPrefix ) // prefix
to append to the stored procedure names
{
    return new CTPCC_OLEDB( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix );
}

CTPCC_OLEDB::CTPCC_OLEDB (
LPCSTR szServer,
// name of SQL server
LPCSTR szUser,
// user name for login
LPCSTR szPassword,
```

```

// password for login
LPCSTR szHost,
// not used
LPCSTR szDatabase,
// name of database to use
LPCWSTR szSPPrefix
// prefix to append to the stored procedure
names
)
: m_pIMalloc(NULL)
{
int
iRc;
int
i;
HRESULT
hr;
IDBInitialize* pIDBInitialize =
NULL; // data source interface
IDBProperties* pIDBProperties =
NULL;
ICommandText* pICommandText;
// SQL command

without parameters
wchar_t
szwServer[iMaxNameLen]; //
Unicode string used to convert to BSTR
wchar_t
szwDatabase[iMaxNameLen]; // Unicode string
used to convert to BSTR
wchar_t
szwUser[iMaxNameLen]; //
Unicode string used to convert to BSTR
wchar_t
szwPassword[iMaxNameLen]; // Unicode string
used to convert to BSTR

// Copy stored procedures prefix
wcsncpy(m_szwSPPrefix, szSPPrefix,
sizeof(m_szwSPPrefix)/sizeof(m_szwSPPrefix[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 SetProperty 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              hTpcVersionOutputAccessor;
    // Structure to bind in accessor
    DBBINDING              acOutputDBBinding[nOutputParams];
    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 tpc_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),
&hTpcVersionOutputAccessor,
acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "CheckSPVersion()");
    }

    hr = pICommandText->Execute(NULL, IID_IRowset,
NULL, NULL, (IUnknown **) &pRowset);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eExecute, "CheckSPVersion()");
    }

    // Fetch the result row handle(s)
    hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eGetNextRows, "CheckSPVersion()");
    }

    // Fetch the actual row data by handle
    hr = pRowset->GetData(rghRow,
hTpcVersionOutputAccessor, &db_sp_version);
    if (FAILED(hr))
    {
        ThrowError(pICommandText,
COLEDBERR::eGetData, "CheckSPVersion()");
    }
}

```

```

// Release row(s)
hr = pRowset->Release();

pICommandText->Release();

// Check the retrieved version
if (strcmp(db_sp_version,sVersion)
    throw new
CTPCC_OLEDB_ERR( CTPCC_OLEDB_ERR::ERR_WRON
G_SP_VERSION);
}

void CTPCC_OLEDB::ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR
szLocation)
{
    HRESULT
        hr;
    //char
    szState[6];
    char
    szMsg[SQL_MAX_MESSAGE_LENGTH];
    char
    szTmp[6*SQL_MAX_MESSAGE_LENGTH];
    COLEDBERR
    *pOLEDBErr; // not
allocated until needed (maybe never)
    int
        iLen;
    // Interfaces
    IErrorInfo*    pIErrorInfoAll    = NULL;
    IErrorInfo*    pIErrorInfoRecord = NULL;
    IErrorRecords* pIErrorRecords    = NULL;
    ISupportErrorInfo* pISupportErrorInfo =
NULL;
    ISQLServerErrorInfo* pISQLServerErrorInfo =
NULL;
    ISQLErrorInfo*
    pISQLErrorInfo = NULL;

    // Information used when cannot get custom error
object
    ERRORINFO
    BasicErrorInfo;
    BSTR
    bstrDescription;
    // Number of error records.
    ULONG        nRecs;
    ULONG        nRec;

    // SQL Server error information from
ISQLServerErrorInfo.
    SSERRORINFO*    pSSErrorInfo = NULL;

```

```

OLECHAR*    pSSErrorStrings = NULL;

assert(pObjectWithError != NULL);

pOLEDBErr = new COLEDBERR(szLocation);

pOLEDBErr->m_NativeError = 0;
pOLEDBErr->m_eAction = eAction;
pOLEDBErr->m_bDeadLock = FALSE;

szTmp[0] = 0;

// Only ask for error information if the interface
supports it.
// Note: SQLOLEDB provider supports error interface,
so this check is
// for good style only.
hr = pObjectWithError-
>QueryInterface(IID_ISupportErrorInfo, (void**)
&pISupportErrorInfo);
if (FAILED(hr))
{
    _snprintf(szMsg, sizeof(szMsg),
"SupportErrorInfo interface not supported (hr=0x%X)", hr);
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
    throw pOLEDBErr;
}
/*if (FAILED(pISupportErrorInfo-
>InterfaceSupportsErrorInfo(IID_InterfaceWithError))
{
    _snprintf(szMsg, sizeof(szMsg),
"InterfaceWithError interface not supported");
    pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr,
szMsg);
    return;
}*/

// Do not test the return of GetErrorInfo. It can
succeed and return
// a NULL pointer in pIErrorInfoAll. Simply test the
pointer.
GetErrorInfo(0, &pIErrorInfoAll);

if (pIErrorInfoAll != NULL)
{
    // Test to see if it's a valid OLE DB
IErrorInfo interface
    // exposing a list of records.

```

```

if (SUCCEEDED(pIErrorInfoAll-
>QueryInterface(IID_IErrorRecords, (void**)
&pIErrorRecords))
{
    pIErrorRecords-
>GetRecordCount(&nRecs);

    // Within each record, retrieve
information from each
// of the defined interfaces.
for (nRec = 0; nRec < nRecs;
nRec++)
{
    // Request the
generic SQL error interface.
    pIErrorRecords-
>GetCustomErrorObject(nRec,
IID_ISQLErrorInfo, // generic SQL error interface
(IUnknown**) &pISQLErrorInfo);
if
(pISQLErrorInfo != NULL)
{
    //
Request SQL Server-specific error interface, not the generic SQL
error interface.
    pISQLErrorInfo->QueryInterface(
IID_ISQLServerErrorInfo, // SQL Server error
interface
    (void**) &pISQLServerErrorInfo);
// Test to ensure the
reference is valid, then
// get error
information from ISQLServerErrorInfo.
if
(pISQLServerErrorInfo != NULL)
{
    pISQLServerErrorInfo->GetErrorInfo(&pSSErrorInfo,
&pSSErrorStrings);
//
ISQLServerErrorInfo::GetErrorInfo succeeds
// even
when it has nothing to return. Test the
//

```

```

pointers before using.
(pSSErrorInfo)
    if
    {
        // 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
    {
        //

```

<pre> Custom error object is not supported. // Use general OLE-DB error interface. // Get the numeric error code pErrorRecords->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. pErrorRecords->GetErrorInfo(nRec, LOCALE_USER_DEFAULT, (IErrorInfo**)&pErrorInfoRecord); if (pErrorInfoRecord) { pErrorInfoRecord- >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); } } </pre>	<pre> } // for() } // if (SUCCEEDED(pErrorInfoAll- >QueryInterface(IID_IErrorRecords, (void**) &pErrorRecords))) 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); } pErrorInfoAll->Release(); } // if (pErrorInfoAll != 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 </pre>	<pre> 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; </pre>
--	---	--

```

        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_RETRIE
D_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitNewOrderParams()
{
    int
        i, j, iOICount;
    HRESULT
        hr;
    wchar_t
        szName[MAX_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
%stpc_neworder (?,?,?,?", m_szSPPrefix);

    // Now print the variable portion
depending on the number of order line parameters
for (iOICount = 0; iOICount <= j;
++iOICount)
    {
        i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"?,?,"");
    }

    // Print the fixed end
if (j != MAX_OL_NEW_ORDER_ITEMS
- 1)
    {
        // append 'default' for the
parameters that are not used
        i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L",default);");
    }
    else // using all 15 order line
parameters
    {
        i += _snwprintf(&szName[i],
sizeof(szName)/sizeof(szName[0]) - i, L"");");
    }

    // Create and Prepare a new command
object for NewOrder.
    CreateCommand(szName,
&m_pINewOrderCommand[j]);

    // Now create the input accessor for this
prepared command
    hr = m_pINewOrderCommand[j]-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
if (FAILED(hr))

```

```

{
    ThrowError(m_pINewOrderCommand[j],
COLEDBERR::eQueryInterface, "InitNewOrderParams()");
}

hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
5 + 3 * (j
+ 1),
acInputDBBinding,
sizeof(NEW_ORDER_DATA),
&m_hNewOrderInputAccessor[j],
acInputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitNewOrderParams()");
}

m_NewOrderExecuteParams[j].cParamSets = 1;
// m_NewOrderExecuteParams.hAccessor
is set dynamically at run-time
// based on the number of new order items
for the particular transaction call.
m_NewOrderExecuteParams[j].hAccessor
= m_hNewOrderInputAccessor[j];
m_NewOrderExecuteParams[j].pData =
&m_txn.NewOrder;

// Create accessor for the first rowset
hr = pIAccessor->CreateAccessor(
DBACCESSOR_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
        IRowsAffected;    // the number of affected rows
for a rowset
        int
        iHandleIndex;    // index into the handle arrays
based on the orders count

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

        iHandleIndex = m_txn.NewOrder.o_ol_cnt - 1;
// for convenience

        while (TRUE)
        {
                try
                {
                        // Execute the prepared
command (according to the number of new orders)
                        // Ask for IMultipleResults
because it returns 2 rowsets.
                        hr =
m_pINewOrderCommand[iHandleIndex]->Execute(

                        NULL, IID_IMultipleResults,

                        &m_NewOrderExecuteParams[iHandleIndex],

                        NULL,

                        (IUnknown **)&pMultipleResults);
                        if (FAILED(hr))
                        {
                                ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eExecute, "NewOrder()");
                        }
                }
        }

```

```

        //////////////////////////////////////
// Get order line results
        //////////////////////////////////////
        m_txn.NewOrder.total_amount
= 0;
        for (i = 0; i <
m_txn.NewOrder.o_ol_cnt; ++i)
        {
                // Get the first
rowset object
                        hr =
pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&IRowsAffected, (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.NewOr
der.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, &IRowsAffected,
(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
        delete e;
        Sleep(10 * iTryCount);
    }
}

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIE
D_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitPaymentParams()
{
    int
        i;
    HRESULT

```

```

        hr;
        wchar_t
        szName[iMAX_SP_NAME_LEN];
        IAccessor*
        pIAccessor;
    const ULONG
        nInputParams = 7; // input parameters
    const ULONG
        nOutputParams = 27; // output result set columns
    // Structure to bind in accessor
    DBBINDING
        acInputDBBinding[nInputParams];
    DBBINDSTATUS
        acInputDBBindStatus[nInputParams];
    DBBINDING
        acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
        acOutputDBBindStatus[nOutputParams];

    // Set command text
    _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]), L"call
%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_INVALI
D_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_RETRIE
D_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitOrderStatusParams()
{
    int
        i;
    HRESULT
        hr;
    wchar_t
        szName[iMAX_SP_NAME_LEN];
    IAccessor*
        pIAccessor;
    const ULONG
        nInputParams = 4; // input parameters
        const ULONG
        nOutputParams = 5; // output 1st result set columns
        const ULONG
        nOutputParams2 = 8; // output 2nd result set columns
    // Structure to bind in accessor
    DBBINDING
        acInputDBBinding[nInputParams];
    DBBINDSTATUS
        acInputDBBindStatus[nInputParams];
    DBBINDING
        acOutputDBBinding[nOutputParams];
    DBBINDSTATUS
        acOutputDBBindStatus[nOutputParams];
    DBBINDING
        acOutputDBBinding2[nOutputParams2];
    DBBINDSTATUS
        acOutputDBBindStatus2[nOutputParams2];
}

```

```

// Set command text
    _snwprintf(szName,
sizeof(szName)/sizeof(szName[0]),
L"{call %stpcc_orderstatus
(?,?,?)}", m_szSPPrefix);

    // Create and Prepare a new command object for
OrderStatus.
    CreateCommand(szName,
&m_pIOrderStatusCommand);

    // Describe the consumer buffer by filling in the array
// of DBBINDING structures. Each binding associates
// a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

    i = 0;
    // OrderStatus parameter 1
    SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, w_id),
sizeof(m_txn.OrderStatus.w_id), DBTYPE_I4);

    // OrderStatus parameter 2
    SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, d_id),
sizeof(m_txn.OrderStatus.d_id), DBTYPE_UI1);

    // OrderStatus parameter 3
    SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

    // OrderStatus parameter 4
    SetBinding(&acInputDBBinding[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

    hr = m_pIOrderStatusCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
    if (FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eQueryInterface, "InitOrderStatusParams()");
    }

    hr = pIAccessor->CreateAccessor(
DBACCESSOR_PARAMETERDATA,
nInputParams,
acInputDBBinding,
sizeof(ORDER_STATUS_DATA),
&m_hOrderStatusInputAccessor,
acInputDBBindStatus);
}

```

```

if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
}

m_OrderStatusExecuteParams.cParamSets = 1;
m_OrderStatusExecuteParams.hAccessor =
m_hOrderStatusInputAccessor;
m_OrderStatusExecuteParams.pData =
&m_txn.OrderStatus;

// Now fill the binding information for result set 1
output columns
InitBindings(&acOutputDBBinding[0],
nOutputParams, eOutputColumn);

// Binding for a rowset that may return more than one
row.
// Bind to offsets of the
OL_ORDER_STATUS_DATA structure instead of
ORDER_STATUS_DATA.
// IRowset::GetData() will be passed individual array
slots OL[i] to fetch the data
// from the row set.

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_supply_w_id),
sizeof(m_txn.OrderStatus.OL[0].ol_supply_w_id), DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_i_id),
sizeof(m_txn.OrderStatus.OL[0].ol_i_id), DBTYPE_I4);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_quantity),
sizeof(m_txn.OrderStatus.OL[0].ol_quantity), DBTYPE_I2);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_amount),
sizeof(m_txn.OrderStatus.OL[0].ol_amount), DBTYPE_R8);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding[i++],
offsetof(OL_ORDER_STATUS_DATA, ol_delivery_d),
sizeof(m_txn.OrderStatus.OL[0].ol_delivery_d),
DBTYPE_DBTIMESTAMP);

hr = pIAccessor->CreateAccessor(

```

```

DBACCESSOR_ROWDATA |
DBACCESSOR_OPTIMIZED,
nOutputParams,
acOutputDBBinding,
sizeof(OL_ORDER_STATUS_DATA),
&m_hOrderStatusOutputAccessor,
acOutputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
}

// Now fill the binding information for result set 2
output columns
InitBindings(&acOutputDBBinding2[0],
nOutputParams2, eOutputColumn);

i = 0;
// OrderStatus output column 1
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_id),
sizeof(m_txn.OrderStatus.c_id), DBTYPE_I4);

// OrderStatus output column 2
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_last),
sizeof(m_txn.OrderStatus.c_last), DBTYPE_STR);

// OrderStatus output column 3
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_first),
sizeof(m_txn.OrderStatus.c_first), DBTYPE_STR);

// OrderStatus output column 4
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_middle),
sizeof(m_txn.OrderStatus.c_middle), DBTYPE_STR);

// OrderStatus output column 5
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_entry_d),
sizeof(m_txn.OrderStatus.o_entry_d),
DBTYPE_DBTIMESTAMP);

// OrderStatus output column 7
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_carrier_id),
sizeof(m_txn.OrderStatus.o_carrier_id), DBTYPE_I2);

// OrderStatus output column 8
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, c_balance),
sizeof(m_txn.OrderStatus.c_balance), DBTYPE_R8);

```

```

// OrderStatus output column 9
SetBinding(&acOutputDBBinding2[i++],
offsetof(ORDER_STATUS_DATA, o_id),
sizeof(m_txn.OrderStatus.o_id), DBTYPE_I4);

hr = pIAccessor->CreateAccessor(
DBACCESSOR_ROWDATA, // cannot be
optimized too because #1 accessor is
nOutputParams2,
acOutputDBBinding2,
sizeof(NEW_ORDER_DATA),
&m_hOrderStatusOutputAccessor2,
acOutputDBBindStatus2);
if (FAILED(hr))
{
    ThrowError(pIAccessor,
COLEDBERR::eCreateAccessor, "InitOrderStatusParams()");
}

void CTPCC_OLEDB::OrderStatus()
{
    HRESULT
    hr;
    int
    iTryCount = 0;
    IMultipleResults* pMultipleResults;
    IRowset* pRowset;
    IRowset* pRowset2;
    LONG cRows =
MAX_OL_ORDER_STATUS_ITEMS; // number of rows
returned in the 1st rowset
    ULONG
    cRowsObtained;
    HROW
    rghRows[MAX_OL_ORDER_STATUS_ITEMS];
//returned row handles for the 1st result set
    HROW*
    prghRows = &rghRows[0];
    LONG cRows2
= 1; // number of rows returned in the 2nd rowset
    ULONG
    cRowsObtained2;
    HROW
    rghRows2; //returned row handle for the
2nd result set
    HROW*
    prghRows2 = &rghRows2;
    int
    i;
    long
    lRowsAffected; // the number of affected rows
for a rowset

```

```

if (m_txn.OrderStatus.c_id != 0)
    m_txn.OrderStatus.c_last[0] = 0;

while (TRUE)
{
    try
    {
        // Execute the prepared
        command
        // Ask for IMultipleResults
        because it returns 2 rowsets.
        hr =
        m_pIOrderStatusCommand->Execute(NULL,
        IID_IMultipleResults, &m_OrderStatusExecuteParams, NULL,

        (IUnknown **)&pMultipleResults);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusCommand,
            COLEDBERR::eExecute, "OrderStatus()");
        }

        // Get order line results
        // Get the first rowset object
        hr = pMultipleResults-
        >GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
        (IUnknown **)&pRowset2);
        if (FAILED(hr))
        {
            ThrowError(m_pIOrderStatusCommand,
            COLEDBERR::eGetResult, "OrderStatus()");
        }

        // Fetch the result row handle(s)
        hr = pRowset2-
        >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 = pRowset2-
>GetData(rghRows[i], m_hOrderStatusOutputAccessor,
&m_txn.OrderStatus.OL[i]);
if (FAILED(hr))
{
    ThrowError(m_pIOrderStatusCommand,
    COLEDBERR::eGetData, "OrderStatus()");
}

// Release row(s)
hr = pRowset2-
>ReleaseRows(cRowsObtained, prghRows, NULL, NULL,
NULL);

// Release rowset
hr = pRowset2->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_SU
CH_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_INVALI
D_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_RETRIE
D_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitDeliveryParams()
{
    int
        i;
    HRESULT
        hr;
    wchar_t
        szName[MAX_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 %stpcc_delivery
(?,?)", m_szSPPrefix);

    // Create and Prepare a new command object for
    Delivery.
        CreateCommand(szName, &m_pIDeliveryCommand);

    // Describe the consumer buffer by filling in the array
    // of DBBINDING structures. Each binding associates
    // a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0], nInputParams,
eInputParameter);

    i = 0;
    // Delivery parameter 1
        SetBinding(&acInputDBBinding[i++],
offsetof(DELIVERY_DATA, w_id),
sizeof(m_txn.Delivery.w_id), DBTYPE_I4);

    // Delivery parameter 2
        SetBinding(&acInputDBBinding[i++],

```

```

offsetof(DELIVERY_DATA, o_carrier_id),
sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);

        hr = m_pIDeliveryCommand-
>QueryInterface(IID_IAccessor, (void **)&pIAccessor);
        if (FAILED(hr))
        {
            ThrowError(m_pIDeliveryCommand,
COLEDBERR::eQueryInterface, "InitDeliveryParams()");
        }

        hr = pIAccessor->CreateAccessor(
            DBACCESSOR_PARAMETERDATA,
            nInputParams,
            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
        rghRow;
    //returned row handles
    HROW*
        prghRow =
&rghRow;

    while (TRUE)
    {
        try
        {
            // Execute the prepared
            command
                hr = m_pIDeliveryCommand-
>Execute(NULL, IID_IRowset, &m_DeliveryExecuteParams,
NULL,
                (IUnknown **)&pRowset);
            if (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eExecute, "Delivery()");
            }

            // Fetch the result row handle(s)
            hr = pRowset-
>GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRow);
            if (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand,
COLEDBERR::eGetNextRows, "Delivery()");
            }

            // Fetch the actual row data by
            handle
                hr = pRowset-
>GetData(rghRow, m_hDeliveryOutputAccessor,
&m_txn.Delivery);

```

```

        if (FAILED(hr))
        {
            ThrowError(m_pIDeliveryCommand,
                COLEDBERR::eGetData, "Delivery()");
        }

        // Release row(s)
        hr = pRowset-
>ReleaseRows(cRowsObtained, prghRow, NULL, NULL,
NULL);

        // Release rowset
        hr = pRowset->Release();

        m_txn.Delivery.exec_status_code = eOK;

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

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

// if (iTryCount)
// throw new
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIE
D_TRANS, iTryCount);
}

```

tpcc_oledb.h

```

/* FILE: TPCC_OLEDB.H
 * Microsoft TPC-C
Kit Ver. 4.20.000
 * Copyright Microsoft,
1999-2004
 * Written by Sergey
Vasilevskiy
 * All Rights Reserved
 *
 * PURPOSE: Header file for TPC-C txn class
OLE DB implementation.

```

```

 *
 *
 */
#pragma once

// need to declare functions for import, unless define has already
// been created
// by the DLL's .cpp module for export.
#ifdef 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_erno = iErr; m_iTryCount = 0; };

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

    int m_erno;
    int m_iTryCount;

    int ErrorType() {return
ERR_TYPE_TPCC_OLEDB};
};

```

```

char*   ErrorTypeStr() { return "TPCC
OLEDB"; }
int     ErrorNum() {return
m_erno;};

char*   ErrorText();
};

class DIIDecl 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_ORD
ER_ITEMS];
// accessor to bind output columns of the
first rowset
HACCESSOR
m_hNewOrderOutputAccessor[MAX_OL_NEW_OR
DER_ITEMS];
// accessor to bind output columns of the
second rowset
HACCESSOR
m_hNewOrderOutputAccessor2[MAX_OL_NEW_O
RDER_ITEMS];
// parameter structure for Execute
DBPARAMS
m_NewOrderExecuteParams[MAX_OL_NEW_ORD
ER_ITEMS];

// Payment
ICommandText*
m_pIPaymentCommand;
HACCESSOR
m_hPaymentInputAccessor; //
accessor to bind input parameters
HACCESSOR
m_hPaymentOutputAccessor; //
accessor to bind output columns
DBPARAMS
m_PaymentExecuteParams;
// parameter structure for Execute

// OrderStatus
ICommandText*
m_pIOrderStatusCommand;
HACCESSOR
m_hOrderStatusInputAccessor; //
accessor to bind input parameters
HACCESSOR
m_hOrderStatusOutputAccessor; //
accessor to bind output columns
HACCESSOR
m_hOrderStatusOutputAccessor2;
// accessor to bind output columns
DBPARAMS

```

```

m_OrderStatusExecuteParams;
// parameter structure for Execute

// Delivery
ICommandText*
m_pIDeliveryCommand;
HACCESSOR
m_hDeliveryInputAccessor; //
accessor to bind input parameters
HACCESSOR
m_hDeliveryOutputAccessor; //
accessor to bind output columns
DBPARAMS
m_DeliveryExecuteParams; //
parameter structure for Execute

wchar_t
m_szSPPrefix[32]; // stored procedures
prefix

// new-order specific fields
int
m_no_commit_flag;

void ThrowError( IUnknown*
pObjectWithError, COLEDBERR::ACTION eAction, LPCTSTR
szLocation );

void CheckSPVersion();

void InitNewOrderParams();
void InitPaymentParams();
void InitDeliveryParams();
void InitStockLevelParams();
void InitOrderStatusParams();

// Helper function to create and prepare a
command
void CreateCommand(wchar_t*
szSqlCommand, ICommandText** ppICommandText);
// Helper function to prepare a command
void PrepareCommand(ICommandText*
pICommand);

// Helper function to fill one binding
// Used for both input parameter and
output column bindings
void SetBinding(DBBINDING*
pDBBinding, size_t obValue, size_t cbMaxLen, DBTYPE
wType);

// Helper function to initialize an array of
bindings

```

```

        void InitBindings(DBBINDING*
pDBBindings, int iCount, eBindingType BindingType);

        union
        {
                NEW_ORDER_DATA
NewOrder;
                PAYMENT_DATA
Payment;
                DELIVERY_DATA
Delivery;
                STOCK_LEVEL_DATA
StockLevel;
                ORDER_STATUS_DATA
OrderStatus;
        }
                m_txn;

public:
        CTPCC_OLEDB(LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost,
LPCSTR szDatabase, LPCWSTR szSPPrefix);
        ~CTPCC_OLEDB(void);

        inline PNEW_ORDER_DATA
BuffAddr_NewOrder()
&m_txn.NewOrder;
        inline PPAYMENT_DATA
BuffAddr_Payment()
&m_txn.Payment;
        inline PDELIVERY_DATA
BuffAddr_Delivery()
&m_txn.Delivery;
        inline PSTOCK_LEVEL_DATA
BuffAddr_StockLevel()
&m_txn.StockLevel;
        inline PORDER_STATUS_DATA
BuffAddr_OrderStatus()
&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, LPCWSTR);

```

dllldata.c

```

/*****
*****
DIIData 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
/dllldata 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 dllldata 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;
        //error id of message
        char szMsg[256];
        //message to sent to browser
} SERRORMSG;

typedef enum _ErrorLevel
{
        ERR_FATAL_LEVEL = 1,
        ERR_WARNING_LEVEL = 2,
        ERR_INFORMATION_LEVEL = 3
} ErrorLevel;

#define ERR_TYPE_LOGIC -1
//logic error in program; internal error
#define ERR_SUCCESS 0
//success (a non-error error)
#define ERR_BAD_ITEM_ID 1

```

```

//expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST
    2
//expected delivery post failed
#define ERR_TYPE_WEBDLL
    3
//tpcc web generated error
#define ERR_TYPE_SQL
    4
//sql server generated error
#define ERR_TYPE_DBLIB
    5
//dblib generated error
#define ERR_TYPE_ODBC
    6
//odbc generated error
#define ERR_TYPE_SOCKET
    7
//error on communication socket client rte only
#define ERR_TYPE_DEADLOCK
    8
//dblib and odbc only deadlock condition
#define ERR_TYPE_COM
    9
//error from COM call
#define ERR_TYPE_TUXEDO
    10
//tuxedo error
#define ERR_TYPE_OS
    11
//operating system error
#define ERR_TYPE_MEMORY
    12
//memory allocation error
#define ERR_TYPE_TPCC_ODBC
    13
//error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB
    14
//error from tpcc dblib txn module
#define ERR_TYPE_DELISRV
    15
//delivery server error
#define ERR_TYPE_TXNLOG
    16
//txn log error
#define ERR_TYPE_BCCONN
    17
//Benchcraft connection class
#define ERR_TYPE_TPCC_CONN
    18
//Benchcraft connection class
#define ERR_TYPE_ENCINA
    19

```

```

//Encina error
#define ERR_TYPE_COMPONENT
    20
//error from COM component
#define ERR_TYPE_RTE
    21
//Benchcraft rte
#define ERR_TYPE_AUTOMATION
    22
//Benchcraft automation errors
#define ERR_TYPE_DRIVER
    23
//Driver engine errors
#define ERR_TYPE_RTE_BASE
    24
//Framework errors
#define ERR_BUF_OVERFLOW
    25
//Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP
    26
//HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB
    27
//OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB
    28
//error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN
    50
//Benchcraft connection class
#define ERR_TYPE_TPCW_HTML
    51
//error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER
    52
//error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE
    53
#define ERR_TYPE_TPCW_ENG_OS
    54
#define ERR_TYPE_HTML_RESP
    55
#define ERR_TYPE_TPCW_ODBC
    56
#define ERR_TYPE_SCHANNEL
    57
#define ERR_TYPE_THINK_LIST
    58
//----- end TPC-W -----
#define ERR_TYPE_XML_PROFILE
    59
// TPC-E error types

```

```

#define ERR_TYPE_TPCE_CONN
    60
//TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE
    61
//TPC-E Rte errors
#define ERR_TYPE_TPCE_ENG_BASE
    62
//Tpce
Driver engine errors
#define ERR_TYPE_TPCE_ENG_OS
    63
//Tpce
Driver engine system errors
//define ERR_TYPE_TPCE_MEE_ENG_BASE
    64
//Tpce
MEE Driver engine errors
//define ERR_TYPE_TPCE_MEE_ENG_OS
    65
//Tpce
MEE Driver engine system errors
#define ERR_INS_MEMORY
    "Insufficient Memory to continue."
#define ERR_UNKNOWN
    "Unknown error."
#define ERR_MSG_BUF_SIZE
    512
#define INV_ERROR_CODE
    -1
#define ERR_INS_BUF_OVERFLOW
    "Insufficient Buffer
size to receive HTML pages."

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

    CBaseErr(LPCTSTR szLoc = NULL)
    {
        m_idMsg = GetLastError();
        //take the error code 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,LPTSTR);

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

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

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

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

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

    virtual int
    ErrorCode()
    { return m_eCode; };
    int
    ErrorAction() { return (int)m_eAction; }
};

```

```

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

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

```

IIS6_Config_CMD

```

@ECHO OFF
@cd %SystemRoot%\System32
@cscript %SYSTEMROOT%\SYSTEM32\iisext.vbs /AddFile
%SYSTEMDRIVE%\INETPUB\WWWROOT\tpcc.dll 1 TPCC
0 TPCC DLL > IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/LogType 0
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/ConnectionTimeout 1200
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY IIS6_CONFIG.OUT
IIS6_CONFIG.ERR /i /y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessExecute True
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessRead True
>> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT

```

```

%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AccessScript True
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AuthAnonymous True
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/1/Root/AuthNTLM True
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/MaxProcesses 1
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolIdentityType 0
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/DefaultAppPool/AppPoolIdentityType 0
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartTime 0
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartRequests 0
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i

```

```

/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PeriodicRestartMemory 0
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleTime False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleRequests False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleSchedule False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleMemory False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleSapiUnhealthy False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleOnDemand False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecycleConfigChange False
    >> IIS6_CONFIG.out

```

```

@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/AppPoolRecyclePrivateMemory False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/PingingEnabled False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul
@CSCRIPT
%SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET
W3SVC/AppPools/RapidFailProtection False
    >> IIS6_CONFIG.out
@if %ErrorLevel% GTR 1 XCOPY
IIS6_CONFIG.ERR+IIS6_CONFIG.OUT IIS6_CONFIG.ERR /i
/y /q >nul

```

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)
*
*      SystemRoot\System32
*/

#include <windows.h>
#include <direct.h>
#include <io.h>

```



```

#include <stdlib.h>
#include <tchar.h>
#include <stdio.h>
#include <comctrl.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 iISMajorVersion;
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 CheckWWWService(void);
static BOOL StartWWWService(void);
static BOOL StopWWWService(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, "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
        siSysInfo.dwNumberOfProcessors =
iNumberOfProcessors =
siSysInfo.dwNumberOfProcessors;

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

        wsprintf(szTmp, "Version
%d.%2.2d.%3.3d", 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 )
    {
        switch( LOWORD(wParam) )
        {
            case
IDOK:
                ProcessOK(hwnd, szDllPath, szWindowsPath);
                return TRUE;
            case
IDCANCEL:

```

```

        EndDialog(hwnd, FALSE);
        return TRUE;
        default:
            return FALSE;
        }
    }
    break;
    default:
        break;
}
return FALSE;
}

static void ProcessOK(HWND hwnd, char *szDllPath, char
*szWindowsPath)
{
    int            d;
    HWND          hDlg;
    int            rc;
    BOOL           bSvcRunning;

    char           szFullName[256];
    char           szErrMsg[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 = CheckWWWebService();
    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) occurred 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 occurred 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
occurred 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\Parameters",
0, KEY_READ, &hKey) == ERROR_SUCCESS )
    {
        if ( iIISMajorVersion == 6 )
        {
            // since IIS6 handles the pool
thread parameters differently, we need to fill in the dialog
rather than PoolThreadLimit // with the MaxPoolThreads

            // 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 )

```

```

15;
                                iListenBackLog =
                                RegCloseKey(hKey);
                                }
                                if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
                                "SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters",
                                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\\Parameters",
        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 ( iISMajorVersion == 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\\Parameters",
        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_MSVC71, 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\PathWWWRoot 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 to
SYSTEM32
            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;

    versionDIIMS = 0;
    versionDIILS = 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);
            versionDIIMS = vs-
>dwProductVersionMS;
            versionDIILS = vs-
>dwProductVersionLS;
            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 CheckWWWebService(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 StartWWWebService(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 StartWWWebErr;
    //start Service pending, Check the status until the
service is running.
    if ( ! QueryServiceStatus(schService, &ssStatus) )
        goto StartWWWebErr;
    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 StartWWWebErr;

    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        szErrMsg[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
//

```

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_LICENSE1 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 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

```

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
//
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif

//
// 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_RTREADING
    EDITTEXT
ED_MAXCONNECTION,164,73,34,12,ES_RIGHT |
ES_NUMBER,
    WS_EX_RTREADING
    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_RIG
HT |
    ES_NUMBER,WS_EX_RTREADING
    EDITTEXT
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,240,34,1
2,ES_RIGHT |
    ES_NUMBER,WS_EX_RTREADING
    EDITTEXT
ED_IIS_THREAD_TIMEOUT,164,254,34,12,ES_RIGHT |
ES_NUMBER,
    WS_EX_RTREADING
    EDITTEXT
ED_IIS_LISTEN_BACKLOG,164,268,34,12,ES_RIGHT |
ES_NUMBER,
    WS_EX_RTREADING
    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_BOR
DER,
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
        BOTTOMMARGIN, 33
    END

    IDD_DIALOG4, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 278
        TOPMARGIN, 7
        BOTTOMMARGIN, 195
    END

```

```

    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 application
// icon
// remains consistent on all systems.
IDI_ICON1     ICON        "icon1.ico"
IDI_ICON2     ICON        "icon2.ico"

//
//
// TPCCDLL
//

IDR_TPCCDLL   TPCCDLL
"..\\..\\visapi_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 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

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

//
//
//
//
#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 <tchar.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

```

```

_bstr_t
_bstr_t
_bstrDllPath = szDllPath;
_variant_t vTmp, vKey;
long lActProp,
ICount, ICountCo, ICountItf, ICountMethod;
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(&ICount);
    if (!SUCCEEDED(hr)) goto Error;

    // iterate through applications to delete existing "TPC-
C" application (if any)
    while (ICount > 0)
    {
        hr = pCatalogCollectionApp-
>get_Item(ICount - 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"))
    {
        ICount--;
        continue;
    }
    else
    {
        hr = pCatalogCollectionApp-
>Remove(ICount - 1);
        if (!SUCCEEDED(hr)) goto
Error;
    }

    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;

        ICountMethod--;
    }

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

        pCatalogObjectItf->Release();
        pCatalogObjectItf = NULL;

        ICountItf--;
    }

        pCatalogObjectCo->Release();
        pCatalogObjectCo = NULL;

        ICountCo--;
    }

    // save changes
    hr = pCatalogCollectionCo-
>SaveChanges(&IActProp);
    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;
}

```

isapi_dll_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

```

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 NONINFRINGEMENT.

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 OF 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 découlant 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 données,

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'avance, avisé de l'éventualité de tels dommages. Certains États/juridictions 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 regard. La présente Convention est régie par les lois de la province d'Ontario, Canada.
Chacune des parties ... la présente reconnaît et accepte, par l'intermédiaire de 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
};

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
CCComObjectRootEx<CCComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
    COM_INTERFACE_ENTRY(ITPCC)
    COM_INTERFACE_ENTRY(IObjectControl)
    COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

    CTPCC_Common();
    ~CTPCC_Common();

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

    HRESULT __stdcall CallSetComplete();

// IObjectControl
    STDMETHODIMP_(BOOL) CanBePooled() { return
m_bCanBePooled;};
    STDMETHODIMP Activate() { return S_OK;};
    // we don't support COM Services transactions (no
enlistment)
    STDMETHODIMP_(void) Deactivate() { /* nothing
to do */};

// IObjectConstruct
    STDMETHODIMP Construct(IDispatch * pUnk);

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

    struct COM_DATA
    {
        int retval;
        int error;
        union

```

```

{
    NEW_ORDER_DATA
NewOrder;
    PAYMENT_DATA
Payment;
    DELIVERY_DATA
Delivery;
    STOCK_LEVEL_DATA
StockLevel;
    ORDER_STATUS_DATA
OrderStatus;
} u;
};

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

    BEGIN_COM_MAP(CTPCC)
        //COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx<CCComSingleThreadModel>)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
    mon)
    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)
    mon)
    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_ORDERSTATU
S)

BEGIN_COM_MAP(COrderStatus)
//      COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
//      COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//      COM_INTERFACE_ENTRY_CHAIN(CTPCC_Com
mon)
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_Com
mon)
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_Com
mon)
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
Tuxedo class.
*      Contact:      Charles Levine (clevine@microsoft.com)
*
*      Change history:
*                               4.20.000 - first version
*/

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

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

```

```

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

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

pReg->bCOM_SinglePool = FALSE;
size = sizeof(szTmp);
if ( RegQueryValueEx(hKey, "COM_SinglePool", 0,
&type, (BYTE *)&szTmp, &size) == ERROR_SUCCESS )
{
    if ( !strcmp(szTmp, "YES") )
        pReg->bCOM_SinglePool =
TRUE;
}

pReg->dwMaxConnections = 0;
size = sizeof(dwTmp);
if ( ( RegQueryValueEx(hKey, "MaxConnections", 0,
&type, (LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
    && (type == REG_DWORD) )
    pReg->dwMaxConnections = dwTmp;

pReg->dwMaxPendingDeliveries = 0;
size = sizeof(dwTmp);

```

```

if ( ( RegQueryValueEx(hKey,
"MaxPendingDeliveries", 0, &type, (LPBYTE)&dwTmp, &size)
== ERROR_SUCCESS )
    && (type == REG_DWORD) )
    pReg->dwMaxPendingDeliveries =
dwTmp;

pReg->dwNumberOfDeliveryThreads = 0;
size = sizeof(dwTmp);
if ( ( RegQueryValueEx(hKey,
"NumberOfDeliveryThreads", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
    && (type == REG_DWORD) )
    pReg->dwNumberOfDeliveryThreads =
dwTmp;

size = sizeof( pReg->szPath );
if ( RegQueryValueEx(hKey, "Path", 0, &type,
(BYTE *)&pReg->szPath, &size) != ERROR_SUCCESS )
    pReg->szPath[0] = 0;

size = sizeof( pReg->szDbServer );
if ( RegQueryValueEx(hKey, "DbServer", 0, &type,
(BYTE *)&pReg->szDbServer, &size) != ERROR_SUCCESS )
    pReg->szDbServer[0] = 0;

size = sizeof( pReg->szDbName );
if ( RegQueryValueEx(hKey, "DbName", 0, &type,
(BYTE *)&pReg->szDbName, &size) != ERROR_SUCCESS )
    pReg->szDbName[0] = 0;

size = sizeof( pReg->szDbUser );
if ( RegQueryValueEx(hKey, "DbUser", 0, &type,
(BYTE *)&pReg->szDbUser, &size) != ERROR_SUCCESS )
    pReg->szDbUser[0] = 0;

size = sizeof( pReg->szDbPassword );
if ( RegQueryValueEx(hKey, "DbPassword", 0,
&type, (BYTE *)&pReg->szDbPassword, &size) !=
ERROR_SUCCESS )
    pReg->szDbPassword[0] = 0;

size = sizeof( pReg->szSPPrefix );
if ( RegQueryValueExW(hKey, L"SPPrefix", 0,
&type, (BYTE *)&pReg->szSPPrefix, &size) !=
ERROR_SUCCESS )
    pReg->szSPPrefix[0] = L'\0';

pReg->dwConnectDelay = 0;
size = sizeof(dwTmp);
if ( ( RegQueryValueEx(hKey, "ConnectDelay", 0,
&type, (LPBYTE)&dwTmp, &size) == ERROR_SUCCESS )
    && (type == REG_DWORD) )
    pReg->dwConnectDelay = dwTmp;

```

```

pReg->bCallNoDuplicatesNewOrder = FALSE;
size = sizeof(dwTmp);
if ( ( RegQueryValueEx(hKey,
"CallNoDuplicatesNewOrder", 0, &type, (LPBYTE)&dwTmp,
&size) == ERROR_SUCCESS )
    && (type == REG_DWORD) )
    pReg->bCallNoDuplicatesNewOrder =
dwTmp;

RegCloseKey(hKey);

return FALSE;
}

```

readregistry.h

```

/* FILE: ReadRegistry.h
* Microsoft TPC-C
* Kit Ver. 4.20.000
* Copyright Microsoft,
1999
* All Rights Reserved
* not audited
* PURPOSE: Header for registry related code.
* Change history:
* 4.20.000 - first version
*/

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

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

//This structure defines the data necessary to keep distinct for
each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
    enum DBPROTOCOL eDB_Protocol;
    enum TXNMON eTxnMon;
    BOOL bCOM_SinglePool;
    DWORD dwMaxConnections;
    DWORD dwMaxPendingDeliveries;
    DWORD dwNumberOfDeliveryThreads;
    char szPath[128];
    char szDbServer[32];
}

```

```

char szDbName[32];
char szDbUser[32];
char szDbPassword[32];
wchar_t szSPPrefix[32];          //tpcc_odbc.dll
stored procedures prefix
    DWORD dwConnectDelay;        // delay in ms to use
in pacing connection open and close
    BOOL bCallNoDuplicatesNewOrder;    //
whether to check for non-duplicate item ids and call a different
New Order SP
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA
*pReg );

```

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_LICENSE1       112
#define IDD_DIALOG4        113
#define IDR_TPCCOBJ1       117
#define IDR_TPCCSTUB1      118
#define IDR_DBLIB_DLL       122
#define IDR_ODBC_DLL        123
#define IDR_TUXEDO_APP      124
#define IDR_TUXEDO_DLL      125
#define IDR_COM_DLL         126
#define IDR_COMPS_DLL       127
#define IDR_COMALL_DLL      128
#define IDR_COMTYPLIB_DLL   129
#define IDR_MSVCRT701       130
#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_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_L
ENGLH+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: DIIMain
 *
 * PURPOSE:      This function is the entry point for the
DLL. This implementation is based on the
 *              fact that
DLL_PROCESS_ATTACH is only called from the inet service
once.
 *
 * ARGUMENTS:    HANDLE hModule
                 module handle
 *
                 DWORD
                 ul_reason_for_call    reason for call
 *
                 LPVOID
                 lpReserved            reserved for future
use
 *
 * RETURNS:      BOOL FALSE
                 errors ocured in initialization
 *
                 TRUE                 DLL
successfully initialized
 */

BOOL APIENTRY DIIMain(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
CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

// get
function pointer to wrapper for class constructor

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

```

```

        throw new
CWEBCLNNT_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
CWEBCLNNT_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
CWEBCLNNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
        }
    }

    // Check whether
Service Pack 1 has been installed if
    // running on
Windows Server 2003. The RTM version has
    // 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

```

```

user's SID                NULL, // current

lpszStrings                1, // strings in

data                      0, // no bytes of raw

array of error strings    (LPCTSTR *)lpszStrings, //

                                                                    NULL); // no raw data

                                                                    (VOID)
DeregisterEventSource(hEventSource);
    }
                                                                    }
                                                                    }
(dwNumDeliveryThreads)    if
    {
                                                                    //
Initialize delivery delay critical section
                                                                    //
                                                                    InitializeCriticalSection(&hConnectCriticalSection);
                                                                    // for
deferred delivery txns:
                                                                    //
                                                                    hDoneEvent = CreateEvent( NULL, TRUE /* manual
reset */, FALSE /* initially not signalled */, NULL );
                                                                    InitializeCriticalSection(&DelBuffCriticalSection);
                                                                    hWorkerSemaphore = CreateSemaphore( NULL, 0,
dwDelBuffSize, NULL );
                                                                    dwDelBuffFreeCount = dwDelBuffSize;

                                                                    InitJulianTime(NULL);

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

                // Delete
                //
                delivery delay critical section

                DeleteCriticalSection(&hConnectCriticalSection);
                }

                DeleteCriticalSection(&TermCriticalSection);

                if
                (hLibInstanceTm != NULL)

                    FreeLibrary( hLibInstanceTm );

                hLibInstanceTm =
                NULL;

                if
                (hLibInstanceDb != NULL)

                    FreeLibrary( hLibInstanceDb );

                hLibInstanceDb =
                NULL;

```

```

                Sleep(500);
                break;

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

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

            WriteMessageToEventLog( szMsg );
            delete e;
            TerminateExtension(0);
            return FALSE;
        }
        catch (...)
        {
            WriteMessageToEventLog(TEXT("Unhandled
exception. DLL could not load."));
            TerminateExtension(0);
            return FALSE;
        }

        return TRUE;
    }

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

    BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO
    *pVer)
    {
        pVer->dwExtensionVersion =
        MAKELONG(HSE_VERSION_MINOR,
        HSE_VERSION_MAJOR);

        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 (txnDelilog != NULL)
                {
                    txnDelilog-
>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(txnDeliLog != 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);

pDeliveryData-

```

```

>w_id = delivery.w_id;
pDeliveryData-
>o_carrier_id = delivery.o_carrier_id;
txnDeliRec.w_id =
pDeliveryData->w_id;
txnDeliRec.o_carrier_id = pDeliveryData-
>o_carrier_id;
txnDeliRec.TxnStartT0 =
Get64BitTime(&delivery.queue);

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

GetLocalTime( &trans_end );

//log txn

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

txnDeliRec.o_id[i] = pDeliveryData->o_id[i];
txnDeliRec.DeltaT4
= (int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartT0);

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

if (txnDelilog !=
NULL)

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

WriteMessageToEventLog( szTmp );

// log the error txn
txnDeliRec.TxnStatus = e-
>ErrorType();

if (txnDelilog != NULL)
txnDelilog-
>WriteToLog(&txnDeliRec);

```

```

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

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

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

EnterCriticalSection(&hConnectCriticalSection);

Sleep(Reg.dwConnectDelay);
}

delete pTxn;

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

void ProcessQueryString(EXTENSION_CONTROL_BLOCK
*pECB, int *pCmd, int *pFormId, int *pTermId, int *pSynclId)
{
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
CWEBCLNTErr( ERR_COMMAND_UNDEFINED );
    if ( !strcmp(szCmds[i], szBuffer) )
    {
        *pCmd = i+1;
        break;
    }
}

/* FUNCTION: void WelcomeForm
*/
void WelcomeForm(EXTENSION_CONTROL_BLOCK

```

```

*pECB, char *szBuffer)
{
    char szTmp[1024];

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

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

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

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

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

    "</PRE></font>"

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

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

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

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

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

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

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

    );

    sprintf( szTmp, "Configuration Settings:
<BR><font face='Courier New' color='blue'><PRE>"
    "Txn
Monitor = <B>%s</B><BR>"

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

```

```

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

printf( 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 iTotal;

    EnterCriticalSection(&TermCriticalSection);

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

    LeaveCriticalSection(&TermCriticalSection);

    wsprintf( szBuffer,

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

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

```

```

{
  {
    ERR_COMMAND_UNDEFINED,
    "Command
undefined."
  },
  {
    ERR_D_ID_INVALID,
    "Invalid District ID Must be 1 to 10."
  },
  {
    ERR_DELIVERY_CARRIER_ID_RANGE,
    "Delivery Carrier ID out of
range must be 1 - 10."
  },
  {
    ERR_DELIVERY_CARRIER_INVALID,
    "Delivery Carrier ID invalid must be
numeric 1 - 10."
  },
  {
    ERR_DELIVERY_MISSING_OCD_KEY,
    "Delivery missing Carrier ID key
\"OCD*\"."
  },
  {
    ERR_DELIVERY_THREAD_FAILED,
    "Could not start delivery
worker thread."
  },
  {
    ERR_GETPROCADDR_FAILED,
    "Could not map
proc in DLL. GetProcAddr error. DLL="
  },
  {
    ERR_HTML_ILL_FORMED,
    "Required key field is missing from HTML string."
  },
  {
    ERR_INVALID_SYNC_CONNECTION,
    "Invalid Terminal Sync ID."
  },
  {
    ERR_INVALID_TERMID,
    "Invalid Terminal ID."
  },
  {
    ERR_LOADDLL_FAILED,
    "Load of DLL failed. DLL="
  }
}

```

```

},
{
  ERR_MAX_CONNECTIONS_EXCEEDED,
  "No connections available.
Max Connections is probably too low."
},
{
  ERR_MISSING_REGISTRY_ENTRIES,
  "Required registry entries are missing.
Rerun INSTALL to correct."
},
{
  ERR_NEWORDER_CUSTOMER_INVALID,
  "New Order customer id
invalid data type, range = 1 to 3000."
},
{
  ERR_NEWORDER_CUSTOMER_KEY,
  "New Order missing Customer
key \"CID*\"."
},
{
  ERR_NEWORDER_DISTRICT_INVALID,
  "New Order District ID Invalid
range 1 - 10."
},
{
  ERR_NEWORDER_FORM_MISSING_DID,
  "New Order missing District
key \"DID*\"."
},
{
  ERR_NEWORDER_ITEMID_INVALID,
  "New Order Item Id is wrong data type,
must be numeric."
},
{
  ERR_NEWORDER_ITEMID_RANGE,
  "New Order Item Id is out of
range. Range = 1 to 999999."
},
{
  ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
  "New Order Item_Id field entered without
a corresponding Supp_W."
},
{
  ERR_NEWORDER_MISSING_IID_KEY,
  "New Order missing Item Id key
\"IID*\"."
},
{
  ERR_NEWORDER_MISSING_QTY_KEY,
  "New Order Missing Qty key
\"Qty##*\"."
},
{
  ERR_NEWORDER_MISSING_SUPPW_KEY,
  "New Order missing Supp_W
key \"SP##*\"."
}

```

```

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

```

```

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

```

```

        ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last
Name must be entered."
    },
    {
        ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key
\CID*\."
    },
    {
        ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name
key \CLT*\."
    },
    {
        ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse
key \CWI*\."
    },
    {
        ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \DID*\."
    },
    {
        ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key
\HAM*\."
    },
    {
        ERR_STOCKLEVEL_MISSING_THRESHOLD_K
EY,
        "Stock Level; missing Threshold key \TT*\."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in
the range = 1 - 99."
    },
    {
        ERR_STOCKLEVEL_THRESHOLD_RANGE,
        "Stock Level Threshold out of
range, range must be 1 - 99."
    },
    {
        ERR_VERSION_MISMATCH,
        "Invalid
version field. RTE and Web Client are probably out of sync."
    },
    {
        ERR_W_ID_INVALID,
        "Invalid Warehouse ID."
    },
    {
        0,
        ""
    },

```

```

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

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

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

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

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

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

    ptr0 = ptr; // remember starting point
    // scan string until a terminator (null or &) or a non-

```

```

digit
while( *ptr && *ptr != '&' && isdigit(*ptr) )
    ptr++;

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

*pQueryString = ptr;
return atoi(ptr0);

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

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

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

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

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

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

```

```

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

        LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

        LeaveCriticalSection(&TermCriticalSection);
}

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

```

```

terminal id
*/

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

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

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

                Term.pClientData[iNewTerm].iTickCount =
GetTickCount();
                Term.pClientData[iNewTerm].iSyncId =

```

```

Term.iMasterSyncId++;
        Term.pClientData[iNewTerm].pTxn = NULL;

        LeaveCriticalSection(&TermCriticalSection);
        return iNewTerm;
}

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

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

                EnterCriticalSection(&TermCriticalSection);

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

                LeaveCriticalSection(&TermCriticalSection);
        }
}

/* FUNCTION: MakeErrorForm
*/

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

```



```

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

/* FUNCTION: MakeMainMenuForm
*/

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

```

```

NAME=\"CMD\" VALUE=\"..Delivery..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
        "</FORM><<BODY><<HTML>"
        , MAIN_MENU_FORM, iTermId,
iSyncId);
}

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

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

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

```

```

        strcpy(szForm+c,
            "Stock Level Threshold:
<INPUT NAME=\"TT*\" SIZE=2><BR> <BR>"
            "low stock: </font><BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR></PRE><HR>"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Process\">"
            "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"Menu\">"
            "</FORM><<HTML>" );
    }
    else
    {
        sprintf(szForm+c,
            "Stock Level Threshold:
%2.2d<BR> <BR>"
            "low stock: %3.3d</font>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR></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->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>

```



```

pNewOrderData->OL[i].ol_i_id,
pNewOrderData->OL[i].ol_i_name,
pNewOrderData->OL[i].ol_quantity,
pNewOrderData->OL[i].ol_stock,
pNewOrderData->OL[i].ol_brand_generic,
pNewOrderData->OL[i].ol_i_price,
pNewOrderData->OL[i].ol_amount );
    }
else
{
    c += sprintf(szForm+c,
                "%Disc:<BR>"
                "Order Number:
%8.8d Number of Lines:    W_tax:    D_tax:<BR>"
                " Supp_W Item Id
Item Name      Qty Stock B/G Price  Amount<BR>"
                , pNewOrderData-
>o_id);
    i = 0;
}
strcpy( szForm+c, szBR, (15-i)*5 );
c += (15-i)*5;

if ( bValid )
    c += sprintf(szForm+c,
                "Execution Status: Transaction committed.          Total:
                $%8.2f ",
                pNewOrderData-
>total_amount);
else
    c += sprintf(szForm+c,
                "Execution Status: Item number is not valid.          Total:");
strcpy(szForm+c,
        "<BR></font></PRE><HR>"
        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..NewOrder..">"
        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..Payment..">"
        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..Delivery..">"
        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..Order-Status..">"

```

```

        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..Stock-Level..">"
        "<INPUT TYPE='submit'"
NAME="CMD" VALUE="..Exit..">"
        "</FORM></HTML>"
    );
}
}

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

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

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

    if ( !bInput )
    {
        c += sprintf(szForm+c, "%2.2d-%2.2d-
%4.4d %2.2d:%2.2d:%2.2d",
                    pPaymentData->h_date.day,
                    pPaymentData->h_date.month,
                    pPaymentData->h_date.year,
                    pPaymentData->h_date.hour,
                    pPaymentData->h_date.minute,
                    pPaymentData-
>h_date.second);

```

```

    }
    if ( bInput )
    {
        c += sprintf(szForm+c,
                    "<BR> <BR>Warehouse:
%6.6d"
                    "
                    District:
<INPUT NAME='DID*' SIZE=1><BR> <BR> <BR> <BR>
<BR>"
                    "Customer: <INPUT
NAME='CID*' SIZE=4>"
                    "Cust-Warehouse: <INPUT
NAME='CWI*' SIZE=4> "
                    "Cust-District: <INPUT
NAME='CDI*' SIZE=1><BR>"
                    "Name:          <INPUT
NAME='CLT*' SIZE=16>          Since:<BR>"
                    "
                    Credit:<BR>"
                    "
                    Disc:<BR>"
                    "
                    Phone:<BR> <BR>"
                    "Amount Paid:    $<INPUT
NAME='HAM*' SIZE=7>    New Cust-Balance:<BR>"
                    "Credit Limit:<BR>
<BR>Cust-Data: <BR> <BR> <BR> <BR>
<BR></font></PRE><HR>"
                    "<INPUT TYPE='submit'"
NAME="CMD" VALUE="Process"><INPUT
TYPE="submit" NAME="CMD" VALUE="Menu">"
                    "</BODY></FORM></HTML>"
                );
        Term.pClientData[iTermId].w_id);
    }
else
{
    c += sprintf(szForm+c,
                "<BR> <BR>Warehouse:
%6.6d      District: %2.2d<BR>"
                "%-20s          %-
20s<BR>"
                "%-20s          %-
20s<BR>"
                "%-20s %-2s %5.5s-%4.4s<BR> <BR>"
                "Customer: %4.4d Cust-
Warehouse: %6.6d Cust-District: %2.2d<BR>"
                "Name: %-16s %-2s %-16s
Since: %2.2d-%2.2d-%4.4d<BR>"
                "          %-20s          Credit:
%-2s<BR>"

```

```

Term.pClientData[iTermId].w_id, pPaymentData->d_id
, pPaymentData->w_street_1,
pPaymentData->d_street_1
, pPaymentData->w_street_2,
pPaymentData->d_street_2
, pPaymentData->w_city,
pPaymentData->w_state, pPaymentData->w_zip,
pPaymentData->w_zip+5
, pPaymentData->d_city,
pPaymentData->d_state, pPaymentData->d_zip, pPaymentData-
>d_zip+5
, pPaymentData->c_id,
pPaymentData->c_d_id
, pPaymentData->c_first,
pPaymentData->c_middle, pPaymentData->c_last
, pPaymentData->c_since.day,
pPaymentData->c_since.month, pPaymentData->c_since.year
, pPaymentData->c_street_1,
pPaymentData->c_credit
);
c += sprintf(szForm+c,
" %20s
%%Disc: %5.2f<BR>",
pPaymentData->c_street_2,
100.0*pPaymentData->c_discount);
c += wsprintf(szForm+c,
" %20s %5.5s-
%4.4s Phone: %6.6s-%3.3s-%3.3s-%4.4s<BR> <BR>",
pPaymentData->c_city,
pPaymentData->c_state, pPaymentData->c_zip, pPaymentData-
>c_zip+5,
pPaymentData->c_phone,
pPaymentData->c_phone+6, pPaymentData->c_phone+9,
pPaymentData->c_phone+12 );
c += sprintf(szForm+c,
"Amount Paid: $%7.2f
New Cust-Balance: $%14.2f<BR>"
"Credit Limit: $%13.2f<BR>
<BR>"
, pPaymentData->h_amount,
pPaymentData->c_balance
, pPaymentData->c_credit_lim
);
if ( pPaymentData->c_credit[0] == 'B' &&
pPaymentData->c_credit[1] == 'C' )
c += wsprintf(szForm+c,
"Cust-
Data: %50.50s<BR> %50.50s<BR> %-

```

```

50.50s<BR> %50.50s<BR>",
pPaymentData->c_data, pPaymentData->c_data+50,
pPaymentData->c_data+100, pPaymentData->c_data+150 );
else
strcpy(szForm+c, "Cust-Data:
<BR> <BR> <BR> <BR>");
strcat(szForm, "
<BR></font></PRE><HR>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..NewOrder..'>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..Payment..'>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..Delivery..'>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..Order-Status..'>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..Stock-Level..'>"
" <INPUT TYPE='submit' NAME='CMD'
VALUE='..Exit..'>"
"</BODY></FORM></HTML>");
}
}
/* FUNCTION: MakeOrderStatusForm
*
* COMMENTS: The internal client buffer is created when
the terminal id is assigned and should not
* be freed except
when the client terminal id is no longer needed.
*/
void MakeOrderStatusForm(int iTermId,
ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput,
char *szForm)
{
int i, c;
static char szBR[] = " <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR>";
c = wsprintf(szForm,
"<HTML><HEAD><TITLE>TPC-C
Order-Status</TITLE></HEAD><BODY>"

```

```

" <FORM ACTION='tpcc.dll'
METHOD='GET'>"
" <INPUT TYPE='hidden'
NAME='STATUSID' VALUE='0'>"
" <INPUT TYPE='hidden'
NAME='ERROR' VALUE='0'>"
" <INPUT TYPE='hidden'
NAME='FORMID' VALUE='%d'>"
" <INPUT TYPE='hidden'
NAME='TERMINID' VALUE='%d'>"
" <INPUT TYPE='hidden'
NAME='SYNCID' VALUE='%d'>"
" <PRE><font face='Courier'>
Order-Status<BR>"
"Warehouse: %6.6d ",
ORDER_STATUS_FORM, iTermId,
Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);
if ( bInput )
{
strcpy(szForm+c,
"District: <INPUT
NAME='DID*' SIZE=1><BR>"
"Customer: <INPUT
NAME='CID*' SIZE=4> Name: <INPUT
NAME='CLT*' SIZE=23><BR>"
"Cust-Balance:<BR> <BR>"
"Order-Number: Entry-
Date: Carrier-Number:<BR>"
"Supply-W Item-Id Qty
Amount Delivery-Date<BR> <BR> <BR> <BR>"
" <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR> <BR> <BR></font></PRE>"
"<HR><INPUT
TYPE='submit' NAME='CMD'
VALUE='Process'><INPUT TYPE='submit'
NAME='CMD' VALUE='Menu'>"
"</BODY></FORM></HTML> " );
}
else
{
c += wsprintf(szForm+c,
"District: %2.2d<BR>"
"Customer: %4.4d Name: %-
16s %2s %-16s<BR>",
pOrderStatusData->d_id,
pOrderStatusData->c_id,
pOrderStatusData->c_first,
pOrderStatusData->c_middle, pOrderStatusData->c_last);
c += sprintf(szForm+c, "Cust-Balance:
$%9.2f<BR> <BR>",

```

```

        pOrderStatusData->c_balance);
    c += 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);
    }
    strcpy( szForm+c, szBR, (15-i)*5 );
    c += (15-i)*5;
    strcpy(szForm+c,
        "</font></PRE><HR><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
        "<INPUT TYPE=\"submit\"

```

```

NAME=\"CMD\" VALUE=\"..Payment..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
        "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
        "</BODY></FORM></HTML>" );
    }
/* FUNCTION: MakeDeliveryForm
*
* COMMENTS: The internal client buffer is created when
the terminal id is assigned and should not
* be freed except
when the client terminal id is no longer needed.
*/
void MakeDeliveryForm(int iTermId, DELIVERY_DATA
*pDeliveryData, BOOL bInput, char *szForm)
{
    int c;
    c = 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>
                "Execution Status: %s <BR>
                <BR> <BR> <BR> <BR> <BR> <BR>
                " <BR> <BR> <BR> <BR>
                <BR> <BR> <BR> <BR> </font></PRE>"
                "<HR><INPUT
TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Payment..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Delivery..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                "<INPUT TYPE=\"submit\"
NAME=\"CMD\" VALUE=\"..Exit..\">"
                "</BODY></FORM></HTML>"
                , pDeliveryData->o_carrier_id,
                (pDeliveryData-
>exec_status_code == eOK) ? "Delivery has been queued." :
                "Delivery Post Failed "
                );
        }
    }
/* FUNCTION: ProcessNewOrderForm
*
* PURPOSE: This function gets and validates the input
data from the new order form
* filling in the required input
variables. it then calls the SQLNewOrder
* transaction, constructs the

```

```

output form and writes it back to client
*                               browser.
*/

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

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

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

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

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

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

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

    pPayment = Term.pClientData[iTermId].pTxn-
>BuffAddr_Payment();

```

```

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

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

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

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

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

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

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

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

```

```

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

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

    PDELIVERY_DATA pDelivery;

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

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

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

```

```

>Delivery();

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

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

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

    PSTOCK_LEVEL_DATA    pStockLevel;

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

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

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

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

```

```

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

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

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

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

```

```

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

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

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

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

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

```

```

_SUPPW );

                                GetKeyValue(&ptr, szQty[i],
szTmp, sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
                                if ( szTmp[0] )
                                        throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SU
PPW );
                                }
                                if ( items == 0 )
                                        throw new
CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERE
D );

                                pNewOrderData->o_ol_cnt = items;
}

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

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

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

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

```

```

    bCustIdBlank = FALSE;
    if ( !IsNumeric(szTmp) )
        throw new
CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALI
D );
        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_LO
NG );

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

    GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_HAM_KEY);
    if (!IsDecimal(szTmp))

```

```

        throw new
CWEBCLNT_ERR( ERR_PAYMENT_HAM_INVALID );
        pPaymentData->h_amount = atof(szTmp);
        if ( pPaymentData->h_amount >= 10000.00 ||
pPaymentData->h_amount < 0 )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_HAM_RANGE );
    }

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

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

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

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

        strcpy(pOrderStatusData->c_last, szTmp);
        // pad with spaces so that the client layer
doesn't have to do it
        // before passing parameters to stored
procedure
        iLen = strlen(pOrderStatusData->c_last);
        memset(pOrderStatusData->c_last + iLen,
' ', LAST_NAME_LEN - iLen);
        pOrderStatusData-
>c_last[LAST_NAME_LEN] = 0;

```



```

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

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

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

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

/* FUNCTION: BOOL IsDecimal(char *ptr)
 *
 * PURPOSE:      This function determines if a string is a
 *               non-negative decimal value.
 *               It fails if any characters other than a series of numbers
 *               followed by
 *               a decimal point, another series
 *               of numbers, and a null terminator are present.
 *
 * ARGUMENTS:   char *ptr

```

```

    pointer to string to check.
 *
 * RETURNS:     BOOL FALSE if string
 *               is not a valid non-negative decimal value
 *
 *               TRUE if string is OK
 */

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

    if ( *ptr == 0 )
        return FALSE;

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

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

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

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

```

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_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_K
EY,

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);
```

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

```

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

#ifdef _WIN32
#ifndef AFX_RESOURCE_DLL
defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32
#endif // _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

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

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

```

```

// Generated from the TEXTINCLUDE 3 resource.
//

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

tpcc_com.cpp

/*      FILE:          TPC_C_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
#defineDllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h" //tpckit
transaction header contains definations of structures specific to
TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "..\..\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
__declspec( dllexport ) CTPCC_COM*
CTPCC_COM_new(BOOL bSinglePool)
{
    return new CTPCC_COM(bSinglePool);
}

```

```

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

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

    m_bSinglePool        = bSinglePool;

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

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

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

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

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

        // all txns will use same component
        m_pPayment = m_pNewOrder;
        m_pStockLevel = m_pNewOrder;
        m_pOrderStatus = m_pNewOrder;
    }
    else
    {
        // use different components for each txn

        hr = CoCreateInstance(CLSID_NewOrder,

```

```

NULL, CLSCTX_SERVER, IID_ITPCC, (void
    **)&m_pNewOrder);
        if (FAILED(hr))
            throw new CCOMERR(hr);

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

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

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

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

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

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

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

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

        ReleaseInterface(m_pNewOrder);
        if (!m_bSinglePool)
        {

```

```

            ReleaseInterface(m_pPayment);
            ReleaseInterface(m_pStockLevel);
            ReleaseInterface(m_pOrderStatus);
        }
        CoUninitialize();
    }

    void CTPCC_COM::NewOrder()
    {
        VARIANT          vTxn_out;

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

        if (FAILED(hr) && 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.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class CCOMERR : public CBaseErr
{
private:
    char m_szErrorText[64];

public:
    // use this interface for genuine COM
errors
    CCOMERR( HRESULT hr )
    {
        m_hr = hr;
        m_iErrorType = 0;
        m_iError = 0;
    }

    // use this interface to impersonate a non-
COM error type
    CCOMERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_hr = S_OK;
    }

    int                m_hr;
    int                m_iErrorType;
    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;                PAYMENT_DATA

DELIVERY_DATA        Delivery;

STOCK_LEVEL_DATA     StockLevel;

ORDER_STATUS_DATA    OrderStatus;
        } u;
        } *m_pTxn;

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

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

        void NewOrder                ();
        void Payment                  ();
        void StockLevel              ();
        void OrderStatus              ();
        void Delivery                 ()
{ throw new CCOMERR(E_NOTIMPL); } // not supported
};

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

```

```

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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

tpcc_com_all.cpp

```

/* FILE:                TPCCC_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 "..\..\common\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_L
ENGTH+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_ENTRI
S );

            if (Reg.eDB_Protocol ==
ODBC)
                {
                    strcpy( szDllName,
Reg.szPath );
                    strcat( szDllName,
"tpcc_odbc.dll");
                    hLibInstanceDb =
LoadLibrary( szDllName );
                    if (hLibInstanceDb
== NULL)
                        throw
new CCOMPONENT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                    // get function
pointer to wrapper for class constructor

                    pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                    if
(pCTPCC_ODBC_new == NULL)
                        throw
new CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
                }
            else
                throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );

            if (Reg.dwConnectDelay > 0)
                {

```

```

InitializeCriticalSection(&hConnectCriticalSection);
                }
            }
            else if (dwReason ==
DLL_PROCESS_DETACH)
                _Module.Term();

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

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

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

            (VOID) DeregisterEventSource(hEventSource);
        }
}

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

```



```

        pUnk = NULL;
    }
}

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

char* CCOMPONENT_ERR::ErrorText(void)
{
    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,
          "Required entries missing from registry."
        },
        { ERR_LOADDLL_FAILED,
          "Load of DLL failed. DLL="
        },
        { ERR_GETPROCADDR_FAILED,
          "Could not map proc in DLL.
GetProcAddr error. DLL="
        },
        { ERR_UNKNOWN_DB_PROTOCOL,
          "Unknown database protocol specified in
registry."
        },
        { 0,
          ""
        }
    };

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

    if (m_szTextDetail)
        strcat( szTmp, m_szTextDetail );
}

```

```

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

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

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

CTPCC_Common::~CTPCC_Common()
{
    // Pace connection close for VIA.
    //
    if (Reg.dwConnectDelay > 0)
    {
        EnterCriticalSection(&hConnectCriticalSection);

        Sleep(Reg.dwConnectDelay);

        LeaveCriticalSection(&hConnectCriticalSection);
    }

    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(&hConnectCriticalSection);

            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_TPCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
        exception in CTPCC_Common::NewOrder."););
        pOutData->retval = ERR_TYPE_LOGIC;
        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCOM;
    }
}

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_TPCCOM;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled
        exception in CTPCC_Common::Payment."););
        pOutData->retval = ERR_TYPE_LOGIC;
        pOutData->error = 0;
        m_bCanBePooled = FALSE;
        return E_TPCCOM;
    }
}

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;

#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus
class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus
class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus
class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus
class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus
class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-

```

```

00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif

```

tpcc_com_all.idl

```

/*      FILE:          TPCC.IDL
 *
 *      Microsoft TPC-C
 *      Kit Ver. 4.20.000
 *
 *      Copyright Microsoft,
 *      1999
 *      All Rights Reserved
 *
 *      not yet audited
 *
 *      PURPOSE:       IDL source for TPCC.dll. This
 *      file is processed by the MIDL tool to
 *      produce the type
 *      library (TPCC.tlb) and marshalling code.
 *
 *      Change history:
 *      4.20.000 - first version
 */

```

```

interface TPCC;
interface NewOrder;
interface OrderStatus;
interface Payment;
interface StockLevel;

```

```

import "oidl.idl";
import "ocidl.idl";
import "..\tpcc_com_ps\src\tpcc_com_ps.idl";

```

```

[
    uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
    version(1.0),
    helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{

```

```

importlib("stdole32.tlb");
importlib("stdole2.tlb");

[
    uuid(122A3128-2520-11D3-BA71-
00C04FBFE08B),
    helpstring("All Txns Class")
]
coclass TPCC
{
    [default] interface ITPCC;
};

[
    uuid(975BAABF-84A7-11D2-BA47-
00C04FBFE08B),
    helpstring("NewOrder Class")
]
coclass NewOrder
{
    [default] interface ITPCC;
};

[
    uuid(266836AD-A50D-11D2-BA4E-
00C04FBFE08B),
    helpstring("OrderStatus Class")
]
coclass OrderStatus
{
    [default] interface ITPCC;
};

[
    uuid(CD02F7EF-A4FA-11D2-BA4E-
00C04FBFE08B),
    helpstring("Payment Class")
]
coclass Payment
{
    [default] interface ITPCC;
};

[
    uuid(2668369E-A50D-11D2-BA4E-
00C04FBFE08B),
    helpstring("StockLevel Class")
]
coclass StockLevel
{

```

```

    [default] interface ITPCC;
};
};

```

tpcc_com_all.rc

```

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

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

//
// English (U.S.) resources
//
#ifndef AFX_RESOURCE_DLL ||
defined(AFX_TARGET_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif // _WIN32

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

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

3 TEXTINCLUDE DISCARDABLE
BEGIN
    "1 TYPELIB ""tpcc_com_all.tlb""\r\n"
    "\0"
END

```

```

#endif // APSTUDIO_INVOKED

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all Module\0"
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END
#endif // !_MAC

////////////////////////////////////
//
// REGISTRY
//

IDR_TPCC      REGISTRY DISCARDABLE
"tpcc_com_all.rgs"

```

```

IDR_NEWORDER      REGISTRY DISCARDABLE
"tpcc_com_no.rgs"
IDR_ORDERSTATUS  REGISTRY DISCARDABLE
"tpcc_com_os.rgs"
IDR_PAYMENT      REGISTRY DISCARDABLE
"tpcc_com_pay.rgs"
IDR_STOCKLEVEL   REGISTRY DISCARDABLE
"tpcc_com_sl.rgs"

////////////////////////////////////
//
// String Table
//

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME      "tpcc_com_all"
END

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

#ifdef APSTUDIO_INVOKED
////////////////////////////////////
//
// Generated from the TEXTINCLUDE 3 resource.
//
1 TYPELIB "tpcc_com_all.tlb"

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

tpcc_com_all.rgs

HKCR
{
    TPCC.AllTxns.1 = s 'All Txns Class'
    {
        CLSID = s '{122A3128-2520-11D3-
BA71-00C04FBFE08B}'
    }
    TPCC.AllTxns = s 'TPCC Class'
    {
        CurVer = s 'TPCC.AllTxns.1'
    }
    NoRemove CLSID
    {
        ForceRemove {122A3128-2520-11D3-
BA71-00C04FBFE08B} = s 'TPCC Class'
    }
}

```

```

{
    ProgID = s 'TPCC.AllTxns.1'
    VersionIndependentProgID = s
'TPCC.AllTxns'
    InprocServer32 = s
'%MODULE%'
    {
        val ThreadingModel
= s 'Both'
    }
}

}

}

tpcc_com_all.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 = {l,w1,w2,{b1,b2,b3,b4,b5,b6,b7,b8}}

#endif !_MIDL_USE_GUIDDEF_

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

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

MIDL_DEFINE_GUID(CLSID,
CLSID_NewOrder,0x975BAABF,0x84A7,0x11D2,0xBA,0x47,

```

```

0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E,0x00,0x00,0x00,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x00,0x00,0x00,0x4F,0xBF,0xE0,0x8B);

MIDL_DEFINE_GUID(CLSID,
CLSID_StockLevel,0x2668369E,0xA50D,0x11D2,0xBA,0x4E,0x00,0x00,0x00,0x4F,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( )

#ifdef _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,
LIBID_TPCCLib,0x122A3117,0x2520,0x11D3,0xBA,0x71,0x00,

```

```
0,0xC0,0x4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID,  
CLSID_TPCC,0x122A3128,0x2520,0x11D3,0xBA,0x71,0x00,0  
xC0,0x4F,0xBF,0xE0,0x8B);
```

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

```
MIDL_DEFINE_GUID(CLSID,  
CLSID_OrderStatus,0x266836AD,0xA50D,0x11D2,0xBA,0x4E  
,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);
```

```
MIDL_DEFINE_GUID(CLSID,  
CLSID_Payment,0xCD02F7EF,0xA4FA,0x11D2,0xBA,0x4E,0x  
00,0xC0,0x4F,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_all_resource.h

```
//{{NO_DEPENDENCIES}}  
// Microsoft Developer Studio generated include file.  
// Used by tpcc_com_all.rc  
//  
#define IDS_PROJNAME 100  
#define IDR_TPCC 101  
#define IDR_NEWORDER 102  
#define IDR_ORDERSTATUS 103  
#define IDR_PAYMENT 104  
#define IDR_STOCKLEVEL 105
```

```
// Next default values for new objects  
//
```

```
#ifdef APSTUDIO_INVOKED  
#ifndef APSTUDIO_READONLY_SYMBOLS  
#define _APS_NEXT_RESOURCE_VALUE 202  
#define _APS_NEXT_COMMAND_VALUE 32768  
#define _APS_NEXT_CONTROL_VALUE 201  
#define _APS_NEXT_SYMED_VALUE 106  
#endif  
#endif
```

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
    STDMETHODIMP_(BOOL) CanBePooled() { return
m_bCanBePooled; }
    STDMETHODIMP Activate() { return S_OK; }
    // we don't support COM Services transactions (no
enlistment)
    STDMETHODIMP_(void) Deactivate() { /* nothing
to do */ }

// IObjectConstruct
    STDMETHODIMP Construct(IDispatch * pUnk);

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

    struct COM_DATA
    {
        int retval;
        int error;
        union
        {
            NEW_ORDER_DATA
            NewOrder;
            PAYMENT_DATA
            Payment;
            DELIVERY_DATA
            Delivery;
            STOCK_LEVEL_DATA
            StockLevel;
            ORDER_STATUS_DATA
            OrderStatus;
        }
    };

```

```

};

};

////////////////////////////////////
// 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_Com
mon)
    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_Com
mon)
    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;}
};

```

```

    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_ORDERSTATU
S)

    BEGIN_COM_MAP(COrderStatus)
        // COM_INTERFACE_ENTRY2(IUnknown,
CComObjectRootEx)
        COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
        COM_INTERFACE_ENTRY_CHAIN(CTPCC_Com
mon)
    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_Com
mon)
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_Com
mon)
    END_COM_MAP()

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

```

tpcc_com_no.rgs

```

HKCR
{
    TPCC.NewOrder.1 = s 'NewOrder Class'
    {
        CLSID = s '{975BAABF-84A7-11D2-
BA47-00C04FBFE08B}'
    }
    TPCC.NewOrder = s 'NewOrder Class'
    {
        CurVer = s 'TPCC.NewOrder.1'
    }
    NoRemove CLSID
    {
        ForceRemove {975BAABF-84A7-11D2-
BA47-00C04FBFE08B} = s 'NewOrder Class'
        {
            ProgID = s 'TPCC.NewOrder.1'
            VersionIndependentProgID = s
'TPCC.NewOrder'
        }
    }
    InprocServer32 = s
'%MODULE%'
    {
        val ThreadingModel
= s 'Both'
    }
}

```

tpcc_com_os.rgs

```

HKCR
{
    TPCC.OrderStatus.1 = s 'OrderStatus Class'
    {
        CLSID = s '{266836AD-A50D-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.OrderStatus = s 'OrderStatus Class'
    {
        CurVer = s 'TPCC.OrderStatus.1'
    }
    NoRemove CLSID
    {
        ForceRemove {266836AD-A50D-11D2-
BA4E-00C04FBFE08B} = s 'OrderStatus Class'
        {
            ProgID = s
'TPCC.OrderStatus.1'
            VersionIndependentProgID = s

```

```

'TPCC.OrderStatus'
    InprocServer32 = s
'%MODULE%'
    {
        val ThreadingModel
= s 'Both'
    }
}

```

tpcc_com_pay.rgs

```

HKCR
{
    TPCC.Payment.1 = s 'Payment Class'
    {
        CLSID = s '{CD02F7EF-A4FA-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.Payment = s 'Payment Class'
    {
        CurVer = s 'TPCC.Payment.1'
    }
    NoRemove CLSID
    {
        ForceRemove {CD02F7EF-A4FA-11D2-
BA4E-00C04FBFE08B} = s 'Payment Class'
        {
            ProgID = s 'TPCC.Payment.1'
            VersionIndependentProgID = s
'TPCC.Payment'
        }
    }
    InprocServer32 = s
'%MODULE%'
    {
        val ThreadingModel
= s 'Both'
    }
}

```

tpcc_com_ps.def

```

LIBRARY "tpcc_com_ps"

EXPORTS
    DllGetClassObject PRIVATE
    DllCanUnloadNow PRIVATE
    GetProxyDllInfo PRIVATE
    DllRegisterServer 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, 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( )
```

```
#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_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE
__MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;
```

```
#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__
```

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

```
EXTERN_C const IID IID_ITPCC;
```

```
#if defined(__cplusplus) && !defined(CINTERFACE)
```

```
MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
```

```
ITPCC : public IUnknown
{
public:
virtual HRESULT 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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE 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 STDMETHODCALLTYPE NewOrder
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE Payment
    (
        [in] VARIANT txn_in,
        [out] VARIANT *txn_out
    );
    HRESULT STDMETHODCALLTYPE Delivery
    (

```

```

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

#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,b
7,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,b
7,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,0x47,0x00,0x
C0,0x4F,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,b
7,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,b
7,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,0x47,0x00,0x
C0,0x4F,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*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__

```

```

#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1023
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,
0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN32__)
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub
because it uses these features:
#error -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, /*/
        /* 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,
{
/* 2 */ NdrFcShort( 0x0 ), /* 0 */
0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3ca ), /* Offset= 970 (974) */
/* 6 */ 0x2b, /*
FC_NON_ENCAPSULATED_UNION */
0x9, /*
FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /* */
/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* 22 */ NdrFcShort( 0x800b ), /* Simple arm type:
FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 28 */ NdrFcShort( 0x8008 ), /* Simple arm type:
FC_LONG */
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* 34 */ NdrFcShort( 0x8001 ), /* Simple arm type:
FC_BYTE */
/* 36 */ NdrFcLong( 0x2 ), /* 2 */
/* 40 */ NdrFcShort( 0x8006 ), /* Simple arm type:
FC_SHORT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x800a ), /* Simple arm type:
FC_FLOAT */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 54 */ 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( 0x200 ), /* 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 */
0x15, /*
FC_STRUCT */
0x7, /* 7 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 308 */
0x12, 0x0, /* FC_UP */
/* 310 */ NdrFcShort( 0xc ), /* Offset= 12 (322) */
/* 312 */
0x1b, /*
FC_CARRAY */
0x1, /* 1 */
/* 314 */ NdrFcShort( 0x2 ), /* 2 */
/* 316 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /*
/* 318 */ NdrFcShort( 0xffff ), /* -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( 0xffff ), /* Offset= -1 (441) */
/* 444 */

FC_CARRAY */
0x1b, /*
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( 0xffff6e ), /* 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 */
0x36, /*
FC_POINTER */
/* 522 */ 0x5c, /* FC_PAD */

```

```

0x5b, /*
FC_END */
/* 524 */

0x11, 0x0, /* FC_RP */
/* 526 */ NdrFcShort( 0xffe0 ), /* Offset= -32 (494) */
/* 528 */

0x21, /*
FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /* */
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 542 */ NdrFcShort( 0xff40 ), /* Offset= -192 (350) */
/* 544 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 546 */

0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8, /* FC_LONG */
0x36, /*
FC_POINTER */
/* 556 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 558 */

0x11, 0x0, /* FC_RP */
/* 560 */ NdrFcShort( 0xffe0 ), /* Offset= -32 (528) */
/* 562 */

0x1b, /*
FC_CARRAY */
0x3, /* 3 */
/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
0x0, /* */
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */

0x4b, /*
FC_PP */
0x5c, /*
FC_PAD */
/* 572 */

0x48, /*

```

FC_VARIABLE_REPEAT */			0x0,	/* 0 */	/* 680 */ NdrFcShort(0xffd4), /* Offset= -44 (636) */
	0x49,	/*	/* 628 */ NdrFcShort(0x1),	/* 1 */	/* 682 */
FC_FIXED_OFFSET */			/* 630 */ 0x19,	/* Corr desc: field pointer,	0x5b,
/* 574 */ NdrFcShort(0x4),	/* 4 */		FC_ULONG */		/*
/* 576 */ NdrFcShort(0x0),	/* 0 */				FC_END */
/* 578 */ NdrFcShort(0x1),	/* 1 */		/* 632 */ NdrFcShort(0x4),	/* 4 */	0x8,
/* 580 */ NdrFcShort(0x0),	/* 0 */		/* 634 */ 0x1,	/* FC_BYTE */	/*
/* 582 */ NdrFcShort(0x0),	/* 0 */		FC_END */		FC_LONG */
/* 584 */ 0x12, 0x0, /* FC_UP */			/* 636 */		/* 684 */ 0x5c,
/* 586 */ NdrFcShort(0x184), /* Offset= 388 (974) */					/* FC_PAD */
/* 588 */			0x1a,	/*	0x5b,
	0x5b,	/*	FC_BOGUS_STRUCT */		/*
FC_END */					/* 686 */
			/* 638 */ NdrFcShort(0x10),	/* 16 */	0x1a,
			/* 640 */ NdrFcShort(0x0),	/* 0 */	/*
FC_LONG */			/* 642 */ NdrFcShort(0xa),	/* Offset= 10 (652) */	FC_BOGUS_STRUCT */
/* 590 */ 0x5c,	/* FC_PAD */		/* 644 */ 0x8,	/* FC_LONG */	/* 688 */ NdrFcShort(0x8),
	0x5b,	/*			/* 690 */ NdrFcShort(0x0),
FC_END */			FC_LONG */		/* 692 */ NdrFcShort(0x6),
/* 592 */			/* 646 */ 0x4c,	/*	/* 694 */ 0x8,
			FC_EMBEDDED_COMPLEX */		/* FC_LONG */
	0x1a,	/*			0x36,
FC_BOGUS_STRUCT */			/* 648 */ NdrFcShort(0xffd8), /* Offset= -40 (608) */	/* 0 */	FC_POINTER */
			/* 650 */ 0x36,	/* FC_POINTER */	/* 696 */ 0x5c,
/* 594 */ NdrFcShort(0x8),	/* 8 */				/* FC_PAD */
/* 596 */ NdrFcShort(0x0),	/* 0 */		FC_END */		0x5b,
/* 598 */ NdrFcShort(0x6),	/* Offset= 6 (604) */		/* 652 */		/* 698 */
/* 600 */ 0x8,	/* FC_LONG */				0x11, 0x0, /* FC_RP */
	0x36,	/*	/* 654 */ NdrFcShort(0xffe4), /* Offset= -28 (626) */		/* 700 */ NdrFcShort(0xffd4), /* Offset= -44 (656) */
FC_POINTER */			/* 656 */		/* 702 */
/* 602 */ 0x5c,	/* FC_PAD */				0x1d,
	0x5b,	/*	FC_CARRAY */		/*
FC_END */					FC_SMFARRAY */
/* 604 */			/* 658 */ NdrFcShort(0x4),	/* 4 */	0x0,
			/* 660 */ 0x19,	/* Corr desc: field pointer,	/* 0 */
/* 606 */ NdrFcShort(0xffd4), /* Offset= -44 (562) */			FC_ULONG */		/* 704 */ NdrFcShort(0x8),
/* 608 */					/* 706 */ 0x1,
	0x2f,	/* FC_IP			/* FC_BYTE */
/*			/* 662 */ NdrFcShort(0x0),	/* 0 */	0x5b,
	0x5a,	/*	/* 664 */		FC_END */
FC_CONSTANT_IID */					/* 708 */
/* 610 */ NdrFcLong(0x2f),	/* 47 */		0x4b,	/*	0x15,
/* 614 */ NdrFcShort(0x0),	/* 0 */				/*
/* 616 */ NdrFcShort(0x0),	/* 0 */		FC_PP */		/* 710 */ NdrFcShort(0x10),
/* 618 */ 0xc0,	/* 192 */				/* 712 */ 0x8,
	0x0,	/* 0 */	FC_PAD */		0x6,
/* 620 */ 0x0,	/* 0 */		/* 666 */		/*
	0x0,	/* 0 */			FC_SHORT */
/* 622 */ 0x0,	/* 0 */		FC_VARIABLE_REPEAT */		/* 714 */ 0x6,
	0x0,	/* 0 */			/* FC_SHORT */
/* 624 */ 0x0,	/* 0 */		FC_FIXED_OFFSET */		0x4c,
	0x46,	/* 70 */	/* 668 */ NdrFcShort(0x4),	/* 4 */	/*
/* 626 */			/* 670 */ NdrFcShort(0x0),	/* 0 */	FC_EMBEDDED_COMPLEX */
	0x1b,	/*	/* 672 */ NdrFcShort(0x1),	/* 1 */	/* 716 */ 0x0,
FC_CARRAY */			/* 674 */ NdrFcShort(0x0),	/* 0 */	/* 0 */
			/* 676 */ NdrFcShort(0x0),	/* 0 */	NdrFcShort(0xffff), /*
			/* 678 */ 0x12, 0x0, /* FC_UP */		Offset= -15 (702) */
					0x5b,
					/*
					FC_END */
					/* 720 */
					0x1a,
					/*
					FC_BOGUS_STRUCT */
					0x3,
					/* 3 */

/* 722 */ NdrFcShort(0x18), /* 24 */		0x5b, /*	/* 810 */		
/* 724 */ NdrFcShort(0x0), /* 0 */	FC_END */			0x16, /*	
/* 726 */ NdrFcShort(0xa), /* Offset= 10 (736) */	/* 770 */		FC_PSTRUCT */		
/* 728 */ 0x8, /* FC_LONG */	FC_CARRAY */	0x1b, /*		0x3, /* 3 */	
				/* 812 */ NdrFcShort(0x8), /* 8 */	
FC_POINTER */		0x1, /* 1 */		/* 814 */	
/* 730 */ 0x4c, /*	/* 772 */ NdrFcShort(0x2), /* 2 */			0x4b, /*	
FC_EMBEDDED_COMPLEX */	/* 774 */ 0x19, /* Corr desc: field pointer,		FC_PP */		
	FC_ULONG */	0x0, /* *		0x5c, /*	
/* 732 */ NdrFcShort(0xffe8), /* Offset= -24 (708) */			FC_PAD */		
/* 734 */ 0x5c, /* FC_PAD */	/* 776 */ NdrFcShort(0x0), /* 0 */		/* 816 */		
	/* 778 */ 0x6, /* FC_SHORT */	0x5b, /*	FC_NO_REPEAT */		
FC_END */	FC_END */			0x5c, /*	
/* 736 */	/* 780 */		FC_PAD */		
		0x16, /*	/* 818 */ NdrFcShort(0x4), /* 4 */		
/* 738 */ NdrFcShort(0xff0c), /* Offset= -244 (494) */	FC_PSTRUCT */		/* 820 */ NdrFcShort(0x4), /* 4 */		
/* 740 */		0x3, /* 3 */	/* 822 */ 0x12, 0x0, /* FC_UP */		
	/* 782 */ NdrFcShort(0x8), /* 8 */		/* 824 */ NdrFcShort(0xffe8), /* Offset= -24 (800) */		
FC_CARRAY */	/* 784 */	0x4b, /*	/* 826 */		
			0x5b, /*		
/* 742 */ NdrFcShort(0x1), /* 1 */	FC_PP */		FC_END */		
/* 744 */ 0x19, /* Corr desc: field pointer,		0x5c, /*			
FC_ULONG */	FC_PAD */			0x8, /*	
	/* 786 */		FC_LONG */		
/* 746 */ NdrFcShort(0x0), /* 0 */	FC_NO_REPEAT */	0x46, /*	/* 828 */ 0x8, /* FC_LONG */		
/* 748 */ 0x1, /* FC_BYTE */			0x5b, /*		
	FC_PAD */	0x5c, /*	FC_END */		
FC_END */	/* 788 */ NdrFcShort(0x4), /* 4 */		/* 830 */		
/* 750 */	/* 790 */ NdrFcShort(0x4), /* 4 */		FC_CARRAY */		
	/* 792 */ 0x12, 0x0, /* FC_UP */			0x1b, /*	
FC_PSTRUCT */	/* 794 */ NdrFcShort(0xffe8), /* Offset= -24 (770) */				
	/* 796 */	0x5b, /*		0x7, /* 7 */	
/* 752 */ NdrFcShort(0x8), /* 8 */	FC_END */			/* 832 */ NdrFcShort(0x8), /* 8 */	
/* 754 */		0x8, /*		/* 834 */ 0x19, /* Corr desc: field pointer,	
				FC_ULONG */	
FC_PP */	FC_LONG */			0x0, /* *	
	/* 798 */ 0x8, /* FC_LONG */	0x5b, /*		/* 836 */ NdrFcShort(0x0), /* 0 */	
FC_PAD */	FC_END */			/* 838 */ 0xb, /* FC_HYPER */	
/* 756 */	/* 800 */		FC_END */	0x5b, /*	
			/* 840 */		
FC_NO_REPEAT */	FC_CARRAY */	0x1b, /*		0x16, /*	
			FC_PSTRUCT */		
FC_PAD */	/* 802 */ NdrFcShort(0x4), /* 4 */			0x3, /* 3 */	
/* 758 */ NdrFcShort(0x4), /* 4 */	/* 804 */ 0x19, /* Corr desc: field pointer,		/* 842 */ NdrFcShort(0x8), /* 8 */		
/* 760 */ NdrFcShort(0x4), /* 4 */	FC_ULONG */	0x3, /* 3 */	/* 844 */		
/* 762 */ 0x12, 0x0, /* FC_UP */				0x4b, /*	
/* 764 */ NdrFcShort(0xffe8), /* Offset= -24 (740) */		0x0, /* *	FC_PP */		
/* 766 */	/* 806 */ NdrFcShort(0x0), /* 0 */			0x5c, /*	
	/* 808 */ 0x8, /* FC_LONG */	0x5b, /*	FC_PAD */		
FC_END */	FC_END */		/* 846 */		
				0x46, /*	
FC_LONG */			FC_NO_REPEAT */		
/* 768 */ 0x8, /* FC_LONG */				0x5c, /*	

```

FC_PAD */
/* 848 */ NdrFcShort( 0x4 ), /* 4 */
/* 850 */ NdrFcShort( 0x4 ), /* 4 */
/* 852 */ 0x12, 0x0, /* FC_UP */
/* 854 */ NdrFcShort( 0xffe8 ), /* Offset= -24 (830) */
/* 856 */
0x5b, /*
FC_END */
0x8, /*
FC_LONG */
/* 858 */ 0x8, /* FC_LONG */
0x5b, /*
FC_END */
/* 860 */
0x15, /*
FC_STRUCT */
0x3, /* 3 */
/* 862 */ NdrFcShort( 0x8 ), /* 8 */
/* 864 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 866 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 868 */
0x1b, /*
FC_CARRAY */
0x3, /* 3 */
/* 870 */ NdrFcShort( 0x8 ), /* 8 */
/* 872 */ 0x7, /* Corr desc: FC_USHORT */
0x0, /*
/* 874 */ NdrFcShort( 0xffd8 ), /* -40 */
/* 876 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 878 */ NdrFcShort( 0xffee ), /* Offset= -18 (860) */
/* 880 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 882 */
0x1a, /*
FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 884 */ NdrFcShort( 0x28 ), /* 40 */
/* 886 */ NdrFcShort( 0xffee ), /* Offset= -18 (868) */
/* 888 */ NdrFcShort( 0x0 ), /* Offset= 0 (888) */
/* 890 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 892 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 894 */ 0x4c, /*

```

```

FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 896 */ NdrFcShort( 0xfd8 ), /* Offset= -520 (376) */
/* 898 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 900 */
0x12, 0x0, /* FC_UP */
/* 902 */ NdrFcShort( 0xfef6 ), /* Offset= -266 (636) */
/* 904 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 906 */ 0x1, /* FC_BYTE */
0x5c, /*
FC_PAD */
/* 908 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 910 */ 0x6, /* FC_SHORT */
0x5c, /*
FC_PAD */
/* 912 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 914 */ 0x8, /* FC_LONG */
0x5c, /*
FC_PAD */
/* 916 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 918 */ 0xb, /* FC_HYPER */
0x5c, /*
FC_PAD */
/* 920 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 922 */ 0xa, /* FC_FLOAT */
0x5c, /*
FC_PAD */
/* 924 */
0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 926 */ 0xc, /* FC_DOUBLE */
0x5c, /*
FC_PAD */
/* 928 */
0x12, 0x0, /* FC_UP */
/* 930 */ NdrFcShort( 0xfd8c ), /* Offset= -628 (302) */
/* 932 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 934 */ NdrFcShort( 0xfd8e ), /* Offset= -626 (308) */
/* 936 */
0x12, 0x10, /*

```

```

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( 0xfdbe ), /* Offset= -578 (368) */
/* 948 */
0x12, 0x10, /*
FC_UP [pointer_deref] */
/* 950 */ NdrFcShort( 0x2 ), /* Offset= 2 (952) */
/* 952 */
0x12, 0x0, /* FC_UP */
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
0x15, /*
FC_STRUCT */
0x7, /* 7 */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6, /* FC_SHORT */
0x1, /*
FC_BYTE */
/* 962 */ 0x1, /* FC_BYTE */
0x8, /*
FC_LONG */
/* 964 */ 0xb, /* FC_HYPER */
0x5b, /*
FC_END */
/* 966 */
0x12, 0x0, /* FC_UP */
/* 968 */ NdrFcShort( 0xff4 ), /* 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_tpc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00}} */

```

```

/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x
00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0
,0x4F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
0,
34,
68,
102,
136,
170
};

static const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo =
{
&Object_StubDesc,
__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0
};

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
&Object_StubDesc,
__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0,
0,
0};

CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
&ITPCC_ProxyInfo,
&IID_ITPCC,
IUnknown_QueryInterface_Proxy,
IUnknown_AddRef_Proxy,
IUnknown_Release_Proxy ,
(void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
(void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
(void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,

```

```

(void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
(void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
(void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
&IID_ITPCC,
&ITPCC_ServerInfo,
9,
0, /* pure interpreted */
CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
0,
NdrOleAllocate,
NdrOleFree,
0,
0,
0,
0,
0,
0,
0,
__MIDL_TypeFormatString.Format,
1, /* -error bounds_check flag */
0x20000, /* Ndr library version */
0,
0x6000169, /* MIDL Version 6.0.361 */
0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* cs routines */
0, /* proxy/server info */
0 /* Reserved5 */
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
(CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
(CInterfaceStubVtbl *) &_ITPCCStubVtbl,
0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
"ITPCC",
0

```

```

};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int *
pIndex )
{
    if(! _tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
    _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    & _tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
#endif _MSC_VER >= 1200
#pragma warning(pop)
#endif

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

/* 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=Win64 (32b run,appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data

```

```

VC __declspec() decoration level:
__declspec(uuid()), __declspec(selectany),
__declspec(novtable)
DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AMD64)

#pragma warning( disable: 4049 ) /* more than 64k source lines
*/
#ifdef _MSC_VER >= 1200
#pragma warning(push)
#endif

#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*/
#ifndef _REDQ_RPCPROXY_H_VERSION_
#define _REQUIRED_RPCPROXY_H_VERSION_ 475
#endif

#include "rpcproxy.h"
#ifndef _RPCPROXY_H_VERSION_
#error this stub requires an updated version of <rpcproxy.h>
#endif // _RPCPROXY_H_VERSION_

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1003
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

```

```

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,
0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING
_MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING
_MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo;

extern const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#ifdef !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 */
        /* 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, */
/* 192 */ 0xa, /* 3 */
/* 194 */ NdrFcShort( 0x20 ), /* 10 */
/* 196 */ NdrFcShort( 0x7 ), /* Ext
/* 198 */ NdrFcShort( 0x20 ), /* 32 */
/* 200 */ NdrFcShort( 0x0 ), /* 32 */
/* 202 */ NdrFcShort( 0x0 ), /* 0 */
/* 204 */ NdrFcShort( 0x0 ), /* 0 */
/* 206 */ 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 */
/* 220 */ 0x0, /* 0 */

/* Procedure CallSetComplete */

/* 220 */ 0x33, /* FC_AUTO_HANDLE */
/* 222 */ 0x6c, /* Old
Flags: object, Oi2 */
/* 224 */ 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, */
/* 236 */ 0xa, /* 1 */
/* 238 */ 0xa, /* 10 */
/* 240 */ 0xa, /* Ext
Flags: new corr desc, */
/* 242 */ NdrFcShort( 0x0 ), /* 0 */
/* 244 */ NdrFcShort( 0x0 ), /* 0 */
/* 246 */ NdrFcShort( 0x0 ), /* 0 */
/* 248 */ 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 */
/* 252 */ 0x0, /* 0 */

}
};

static const MIDL_TYPE_FORMAT_STRING
__MIDL_TypeFormatString =
{
0,
{
NdrFcShort( 0x0 ), /* 0 */
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( 0xffff ), /* -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 */
/* 22 */ NdrFcShort( 0x800b ), /* Simple arm type:
FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 26 */ NdrFcShort( 0x8008 ), /* Simple arm type:
FC_LONG */
/* 28 */ NdrFcLong( 0x11 ), /* 17 */
/* 30 */ NdrFcShort( 0x8001 ), /* Simple arm type:
FC_BYTE */
/* 32 */ NdrFcLong( 0x2 ), /* 2 */
/* 34 */ NdrFcShort( 0x8006 ), /* Simple arm type:
FC_SHORT */
/* 36 */ NdrFcLong( 0x4 ), /* 4 */
/* 38 */ NdrFcShort( 0x800a ), /* Simple arm type:
FC_FLOAT */
/* 40 */ NdrFcLong( 0x5 ), /* 5 */
/* 42 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 44 */ NdrFcLong( 0xb ), /* 11 */
/* 46 */ NdrFcShort( 0x8006 ), /* Simple arm type:
FC_SHORT */
/* 48 */ NdrFcLong( 0xa ), /* 10 */

```

```

/* 66 */ NdrFcShort( 0x8008 ), /* Simple arm type:
FC_LONG */
/* 68 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /* Offset= 232 (304) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 74 */ NdrFcShort( 0x800c ), /* Simple arm type:
FC_DOUBLE */
/* 76 */ NdrFcLong( 0x8 ), /* 8 */
/* 78 */ NdrFcShort( 0xe2 ), /* Offset= 226 (310) */
/* 80 */ NdrFcLong( 0xd ), /* 13 */
/* 82 */ NdrFcShort( 0xf6 ), /* Offset= 246 (336) */
/* 84 */ NdrFcLong( 0x9 ), /* 9 */
/* 86 */ NdrFcShort( 0x102 ), /* Offset= 258 (354) */
/* 88 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 90 */ NdrFcShort( 0x10e ), /* Offset= 270 (372) */
/* 92 */ NdrFcLong( 0x24 ), /* 36 */
/* 94 */ NdrFcShort( 0x304 ), /* Offset= 772 (880) */
/* 96 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 98 */ NdrFcShort( 0x2fe ), /* Offset= 766 (880) */
/* 100 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 102 */ NdrFcShort( 0x2fc ), /* Offset= 764 (884) */
/* 104 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 106 */ NdrFcShort( 0x2fa ), /* Offset= 762 (888) */
/* 108 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 110 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (892) */
/* 112 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 114 */ NdrFcShort( 0x2f6 ), /* Offset= 758 (896) */
/* 116 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 118 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (900) */
/* 120 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 122 */ NdrFcShort( 0x2fd ), /* Offset= 754 (904) */
/* 124 */ NdrFcLong( 0x400b ), /* 16395 */
/* 126 */ NdrFcShort( 0x2dc ), /* Offset= 732 (888) */
/* 128 */ NdrFcLong( 0x400a ), /* 16394 */
/* 130 */ NdrFcShort( 0x2da ), /* Offset= 730 (892) */
/* 132 */ NdrFcLong( 0x4006 ), /* 16390 */
/* 134 */ NdrFcShort( 0x2e4 ), /* Offset= 740 (908) */
/* 136 */ NdrFcLong( 0x4007 ), /* 16391 */
/* 138 */ NdrFcShort( 0x2da ), /* Offset= 730 (904) */
/* 140 */ NdrFcLong( 0x4008 ), /* 16392 */
/* 142 */ NdrFcShort( 0x2dc ), /* Offset= 732 (912) */
/* 144 */ NdrFcLong( 0x400d ), /* 16397 */
/* 146 */ NdrFcShort( 0x2da ), /* Offset= 730 (916) */
/* 148 */ NdrFcLong( 0x4009 ), /* 16393 */
/* 150 */ NdrFcShort( 0x2d8 ), /* Offset= 728 (920) */
/* 152 */ NdrFcLong( 0x6000 ), /* 24576 */
/* 154 */ NdrFcShort( 0x2d6 ), /* Offset= 726 (924) */
/* 156 */ NdrFcLong( 0x400c ), /* 16396 */
/* 158 */ NdrFcShort( 0x2d4 ), /* Offset= 724 (928) */
/* 160 */ NdrFcLong( 0x10 ), /* 16 */
/* 162 */ NdrFcShort( 0x8002 ), /* Simple arm type:
FC_CHAR */
/* 164 */ NdrFcLong( 0x12 ), /* 18 */
/* 166 */ NdrFcShort( 0x8006 ), /* Simple arm type:

```

FC_SHORT */					
/* 218 */ NdrFcLong(0x13), /* 19 */		/* 324 */ 0x6,	/* FC_SHORT */		/* 380 */
/* 222 */ NdrFcShort(0x8008), /* Simple arm type:		FC_END */	0x5b,	/*	0x2a,
FC_LONG */		/* 326 */			FC_ENCAPSULATED_UNION */
/* 224 */ NdrFcLong(0x15), /* 21 */			0x17,	/*	0x89,
/* 228 */ NdrFcShort(0x800b), /* Simple arm type:		FC_CSTRUCT */		/*	/* 137 */
FC_HYPER */			0x3,	/* 3 */	
/* 230 */ NdrFcLong(0x16), /* 22 */		/* 328 */ NdrFcShort(0x8), /* 8 */			/* 382 */ NdrFcShort(0x20), /* 32 */
/* 234 */ NdrFcShort(0x8008), /* Simple arm type:		/* 330 */ NdrFcShort(0xffff0), /* Offset= -16 (314) */			/* 384 */ NdrFcShort(0xa), /* 10 */
FC_LONG */		/* 332 */ 0x8,	/* FC_LONG */		/* 386 */ NdrFcLong(0x8), /* 8 */
/* 236 */ NdrFcLong(0x17), /* 23 */			0x8,	/*	/* 390 */ NdrFcShort(0x50), /* Offset= 80 (470) */
/* 240 */ NdrFcShort(0x8008), /* Simple arm type:		FC_LONG */			/* 392 */ NdrFcLong(0xd), /* 13 */
FC_LONG */		/* 334 */ 0x5c,	/* FC_PAD */		/* 396 */ NdrFcShort(0x70), /* Offset= 112 (508) */
/* 242 */ NdrFcLong(0xe), /* 14 */			0x5b,	/*	/* 398 */ NdrFcLong(0x9), /* 9 */
/* 246 */ NdrFcShort(0x2b2), /* Offset= 690 (936) */		FC_END */			/* 402 */ NdrFcShort(0x90), /* Offset= 144 (546) */
/* 248 */ NdrFcLong(0x400e), /* 16398 */		/* 336 */			/* 404 */ NdrFcLong(0xc), /* 12 */
/* 252 */ NdrFcShort(0x2b6), /* Offset= 694 (946) */					/* 408 */ NdrFcShort(0xb0), /* Offset= 176 (584) */
/* 254 */ NdrFcLong(0x4010), /* 16400 */		*/	0x2f,	/* FC_IP	/* 410 */ NdrFcLong(0x24), /* 36 */
/* 258 */ NdrFcShort(0x2b4), /* Offset= 692 (950) */				/*	/* 414 */ NdrFcShort(0x102), /* Offset= 258 (672) */
/* 260 */ NdrFcLong(0x4012), /* 16402 */		FC_CONSTANT_IID */	0x5a,	/*	/* 416 */ NdrFcLong(0x800d), /* 32781 */
/* 264 */ NdrFcShort(0x270), /* Offset= 624 (888) */		/* 338 */ NdrFcLong(0x0), /* 0 */			/* 420 */ NdrFcShort(0x11e), /* Offset= 286 (706) */
/* 266 */ NdrFcLong(0x4013), /* 16403 */		/* 342 */ NdrFcShort(0x0), /* 0 */			/* 422 */ NdrFcLong(0x10), /* 16 */
/* 270 */ NdrFcShort(0x26e), /* Offset= 622 (892) */		/* 344 */ NdrFcShort(0x0), /* 0 */			/* 426 */ NdrFcShort(0x138), /* Offset= 312 (738) */
/* 272 */ NdrFcLong(0x4015), /* 16405 */		/* 346 */ 0xc0,	/* 192 */		/* 428 */ NdrFcLong(0x2), /* 2 */
/* 276 */ NdrFcShort(0x26c), /* Offset= 620 (896) */			0x0,	/* 0 */	/* 432 */ NdrFcShort(0x14e), /* Offset= 334 (766) */
/* 278 */ NdrFcLong(0x4016), /* 16406 */		/* 348 */ 0x0,	/* 0 */		/* 434 */ NdrFcLong(0x3), /* 3 */
/* 282 */ NdrFcShort(0x262), /* Offset= 610 (892) */			0x0,	/* 0 */	/* 438 */ NdrFcShort(0x164), /* Offset= 356 (794) */
/* 284 */ NdrFcLong(0x4017), /* 16407 */		/* 350 */ 0x0,	/* 0 */		/* 440 */ NdrFcLong(0x14), /* 20 */
/* 288 */ NdrFcShort(0x25c), /* Offset= 604 (892) */			0x0,	/* 0 */	/* 444 */ NdrFcShort(0x17a), /* Offset= 378 (822) */
/* 290 */ NdrFcLong(0x0), /* 0 */		/* 352 */ 0x0,	/* 0 */		/* 446 */ NdrFcShort(0xffff), /* Offset= -1 (445) */
/* 294 */ NdrFcShort(0x0), /* Offset= 0 (294) */			0x46,	/* 70 */	/* 448 */
/* 296 */ NdrFcLong(0x1), /* 1 */		/* 354 */			FC_BOGUS_ARRAY */
/* 300 */ NdrFcShort(0x0), /* Offset= 0 (300) */			0x2f,	/* FC_IP	0x21,
/* 302 */ NdrFcShort(0xffff), /* Offset= -1 (301) */		*/		/*	0x3,
/* 304 */	0x15,		0x5a,	/*	/* 3 */
FC_STRUCT */		FC_CONSTANT_IID */			/* 450 */ NdrFcShort(0x0), /* 0 */
/* 306 */ NdrFcShort(0x8), /* 8 */	0x7,	/* 356 */ NdrFcLong(0x20400), /* 132096 */			/* 452 */ 0x19,
/* 308 */ 0xb,	/* 7 */	/* 360 */ NdrFcShort(0x0), /* 0 */			/* Corr desc: field pointer,
	/* FC_HYPER */	/* 362 */ NdrFcShort(0x0), /* 0 */			FC_ULONG */
	0x5b,	/* 364 */ 0xc0,	/* 192 */		0x0,
FC_END */			0x0,	/* 0 */	/* 454 */ NdrFcShort(0x0), /* 0 */
/* 310 */		/* 366 */ 0x0,	/* 0 */		/* 456 */ NdrFcShort(0x1), /* Corr flags: early, */
	0x12, 0x0, /* FC_UP */		0x0,	/* 0 */	/* 458 */ NdrFcLong(0xffffffff), /* -1 */
/* 312 */ NdrFcShort(0xe), /* Offset= 14 (326) */		/* 368 */ 0x0,	/* 0 */		/* 462 */ NdrFcShort(0x0), /* Corr flags: */
/* 314 */			0x0,	/* 0 */	/* 464 */
FC_CARRAY */		/* 370 */ 0x0,	/* 0 */		0x12, 0x0, /* FC_UP */
	0x1b,		0x46,	/* 70 */	/* 466 */ NdrFcShort(0xff74), /* Offset= -140 (326) */
/* 316 */ NdrFcShort(0x2), /* 2 */		/* 372 */			/* 468 */ 0x5c,
/* 318 */ 0x9,	/* Corr desc: FC_ULONG */		0x12, 0x10,	/*	/* FC_PAD */
	0x0,	FC_UP [pointer_deref] */			0x5b,
/* 320 */ NdrFcShort(0xfffc), /* -4 */		/* 374 */ NdrFcShort(0x2), /* Offset= 2 (376) */			FC_END */
/* 322 */ NdrFcShort(0x1), /* Corr flags: early, */		/* 376 */	0x12, 0x0, /* FC_UP */		/* 470 */
		/* 378 */ NdrFcShort(0x1e4), /* Offset= 484 (862) */			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 */		/* 540 */ 0x4c,	/*	/* 596 */	
/* 480 */ 0x36,	/* FC_POINTER */	FC_EMBEDDED_COMPLEX */	0x0,	/* 598 */ NdrFcShort(0xffdc), /* Offset= -36 (562) */	0x11, 0x0, /* FC_RP */
	0x5b,		/* 0 */	/* 600 */	
FC_END */		/* 542 */ NdrFcShort(0xff44), /* Offset= -188 (354) */			
/* 482 */		/* 544 */ 0x5c,	/* FC_PAD */		0x2f,
			0x5b,	/*	/* FC_IP
	0x11, 0x0, /* FC_RP */	FC_END */			
/* 484 */ NdrFcShort(0xffdc), /* Offset= -36 (448) */		/* 546 */			0x5a,
/* 486 */					/*
	0x21,	FC_BOGUS_STRUCT */	0x1a,	FC_CONSTANT_IID */	
	/*		/*	/* 602 */ NdrFcLong(0x2f), /* 47 */	
FC_BOGUS_ARRAY */				/* 606 */ NdrFcShort(0x0), /* 0 */	
	0x3,	/* 548 */ NdrFcShort(0x10), /* 16 */		/* 608 */ NdrFcShort(0x0), /* 0 */	
/* 488 */ NdrFcShort(0x0), /* 0 */	/* 3 */	/* 550 */ NdrFcShort(0x0), /* 0 */		/* 610 */ 0xc0,	/* 192 */
/* 490 */ 0x19,	/* Corr desc: field pointer,	/* 552 */ NdrFcShort(0x6), /* Offset= 6 (558) */		/* 612 */ 0x0,	0x0,
FC_ULONG */		/* 554 */ 0x8,	/* FC_LONG */		/* 0 */
	0x0,		0x40,	/* 614 */ 0x0,	0x0,
/* 492 */ NdrFcShort(0x0), /* 0 */	/* */	FC_STRUCTPAD4 */		/* 616 */ 0x0,	0x0,
/* 494 */ NdrFcShort(0x1), /* Corr flags: early, */		/* 556 */ 0x36,	/* FC_POINTER */		/* 0 */
/* 496 */ NdrFcLong(0xffffffff), /* -1 */			0x5b,	/* 618 */	0x46,
/* 500 */ NdrFcShort(0x0), /* Corr flags: */		FC_END */			/* 70 */
/* 502 */ 0x4c,	/*	/* 558 */			0x1b,
FC_EMBEDDED_COMPLEX */				FC_CARRAY */	
	0x0,	/* 560 */ NdrFcShort(0xffdc), /* Offset= -36 (524) */			0x0,
/* 504 */ NdrFcShort(0xff58), /* Offset= -168 (336) */	/* 0 */	/* 562 */		/* 620 */ NdrFcShort(0x1), /* 1 */	/* 0 */
/* 506 */ 0x5c,	/* FC_PAD */			/* 622 */ 0x19,	/* Corr desc: field pointer,
	0x5b,	FC_BOGUS_ARRAY */		FC_ULONG */	
FC_END */					0x0,
/* 508 */		/* 564 */ NdrFcShort(0x0), /* 0 */		/* 624 */ NdrFcShort(0x4), /* 4 */	/* */
	0x1a,	/* 566 */ 0x19,	/* Corr desc: field pointer,	/* 626 */ NdrFcShort(0x1), /* Corr flags: early, */	
FC_BOGUS_STRUCT */		FC_ULONG */		/* 628 */ 0x1,	/* FC_BYTE */
	0x3,				0x5b,
/* 510 */ NdrFcShort(0x10), /* 16 */	/* 3 */	/* 568 */ NdrFcShort(0x0), /* 0 */		FC_END */	
/* 512 */ NdrFcShort(0x0), /* 0 */		/* 570 */ NdrFcShort(0x1), /* Corr flags: early, */		/* 630 */	
/* 514 */ NdrFcShort(0x6), /* Offset= 6 (520) */		/* 572 */ NdrFcLong(0xffffffff), /* -1 */			0x1a,
/* 516 */ 0x8,	/* FC_LONG */	/* 576 */ NdrFcShort(0x0), /* Corr flags: */		FC_BOGUS_STRUCT */	
	0x40,	/* 578 */			0x3,
FC_STRUCTPAD4 */				/* 632 */ NdrFcShort(0x18), /* 24 */	/* 3 */
/* 518 */ 0x36,	/* FC_POINTER */			/* 634 */ NdrFcShort(0x0), /* 0 */	
	0x5b,	/* 580 */ NdrFcShort(0x176), /* Offset= 374 (954) */		/* 636 */ NdrFcShort(0xa), /* Offset= 10 (646) */	
FC_END */		/* 582 */ 0x5c,	/* FC_PAD */	/* 638 */ 0x8,	/* FC_LONG */
/* 520 */			0x5b,		0x8,
	0x11, 0x0, /* FC_RP */	FC_END */		FC_LONG */	
/* 522 */ NdrFcShort(0xffdc), /* Offset= -36 (486) */		/* 584 */		/* 640 */ 0x4c,	/*
/* 524 */				FC_EMBEDDED_COMPLEX */	
	0x21,	FC_BOGUS_STRUCT */	0x1a,		0x0,
	/*		/*	/* 642 */ NdrFcShort(0xffd6), /* Offset= -42 (600) */	/* 0 */
FC_BOGUS_ARRAY */				/* 644 */ 0x36,	/* FC_POINTER */
	0x3,	/* 586 */ NdrFcShort(0x10), /* 16 */			0x5b,
/* 526 */ NdrFcShort(0x0), /* 0 */	/* 3 */	/* 588 */ NdrFcShort(0x0), /* 0 */		FC_END */	
/* 528 */ 0x19,	/* Corr desc: field pointer,	/* 590 */ NdrFcShort(0x6), /* Offset= 6 (596) */		/* 646 */	
FC_ULONG */		/* 592 */ 0x8,	/* FC_LONG */		0x12, 0x0, /* FC_UP */
	0x0,		0x40,	/* 648 */ NdrFcShort(0xffe2), /* Offset= -30 (618) */	/*
/* 530 */ NdrFcShort(0x0), /* 0 */	/* */	FC_STRUCTPAD4 */		/* 650 */	
/* 532 */ NdrFcShort(0x1), /* Corr flags: early, */		/* 594 */ 0x36,	/* FC_POINTER */		0x21,
/* 534 */ NdrFcLong(0xffffffff), /* -1 */			0x5b,		/*
/* 538 */ NdrFcShort(0x0), /* Corr flags: */		FC_END */			

```

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( 0xff1 ), /*
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 */
    0x5b, /*
FC_END */
/* 738 */
    0x1a, /*
FC_BOGUS_STRUCT */
    0x3, /* 3 */
/* 740 */ NdrFcShort( 0x10 ), /* 16 */
/* 742 */ NdrFcShort( 0x0 ), /* 0 */
/* 744 */ NdrFcShort( 0x6 ), /* Offset= 6 (750) */
/* 746 */ 0x8, /* FC_LONG */
    0x40, /*
FC_STRUCTPAD4 */
/* 748 */ 0x36, /* FC_POINTER */
    0x5b, /*
FC_END */
/* 750 */
    0x12, 0x0, /* FC_UP */
/* 752 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (726) */
/* 754 */
    0x1b, /*

```

```

FC_CARRAY */
    0x1, /* 1 */
/* 756 */ NdrFcShort( 0x2 ), /* 2 */
/* 758 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
    0x0, /* */
/* 760 */ NdrFcShort( 0x0 ), /* 0 */
/* 762 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 764 */ 0x6, /* FC_SHORT */
    0x5b, /*
FC_END */
/* 766 */
    0x1a, /*
FC_BOGUS_STRUCT */
    0x3, /* 3 */
/* 768 */ NdrFcShort( 0x10 ), /* 16 */
/* 770 */ NdrFcShort( 0x0 ), /* 0 */
/* 772 */ NdrFcShort( 0x6 ), /* Offset= 6 (778) */
/* 774 */ 0x8, /* FC_LONG */
    0x40, /*
FC_STRUCTPAD4 */
/* 776 */ 0x36, /* FC_POINTER */
    0x5b, /*
FC_END */
/* 778 */
    0x12, 0x0, /* FC_UP */
/* 780 */ NdrFcShort( 0xffe6 ), /* Offset= -26 (754) */
/* 782 */
    0x1b, /*
FC_CARRAY */
    0x3, /* 3 */
/* 784 */ NdrFcShort( 0x4 ), /* 4 */
/* 786 */ 0x19, /* Corr desc: field pointer,
FC_ULONG */
    0x0, /* */
/* 788 */ NdrFcShort( 0x0 ), /* 0 */
/* 790 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 792 */ 0x8, /* FC_LONG */
    0x5b, /*
FC_END */
/* 794 */
    0x1a, /*
FC_BOGUS_STRUCT */
    0x3, /* 3 */
/* 796 */ NdrFcShort( 0x10 ), /* 16 */
/* 798 */ NdrFcShort( 0x0 ), /* 0 */
/* 800 */ NdrFcShort( 0x6 ), /* Offset= 6 (806) */
/* 802 */ 0x8, /* FC_LONG */
    0x40, /*
FC_STRUCTPAD4 */
/* 804 */ 0x36, /* FC_POINTER */
    0x5b, /*
FC_END */
/* 806 */

```

0x12, 0x0, /* FC_UP */	/* 858 */ NdrFcShort(0xffec), /* Offset= -20 (838) */	[simple_pointer] */
/* 808 */ NdrFcShort(0xffe6), /* Offset= -26 (782) */	/* 860 */ 0x5c, /* FC_PAD */	/* 902 */ 0xa, /* FC_FLOAT */
/* 810 */	0x5b, /*	0x5c, /*
FC_CARRAY */	FC_END */	FC_PAD */
0x1b, /*	/* 862 */	/* 904 */
0x7, /* 7 */	0x1a, /*	0x12, 0x8, /* FC_UP
/* 812 */ NdrFcShort(0x8), /* 8 */	FC_BOGUS_STRUCT */	[simple_pointer] */
/* 814 */ 0x19, /* Corr desc: field pointer,	0x3, /* 3 */	/* 906 */ 0xc, /* FC_DOUBLE */
FC_ULONG */	/* 864 */ NdrFcShort(0x38), /* 56 */	0x5c, /*
0x0, /* */	/* 866 */ NdrFcShort(0xffec), /* Offset= -20 (846) */	FC_PAD */
/* 816 */ NdrFcShort(0x0), /* 0 */	/* 868 */ NdrFcShort(0x0), /* Offset= 0 (868) */	/* 908 */
/* 818 */ NdrFcShort(0x1), /* Corr flags: early, */	/* 870 */ 0x6, /* FC_SHORT */	0x12, 0x0, /* FC_UP */
/* 820 */ 0xb, /* FC_HYPER */	0x6, /*	/* 910 */ NdrFcShort(0xfda2), /* Offset= -606 (304) */
0x5b, /*	FC_SHORT */	/* 912 */
FC_END */	/* 872 */ 0x8, /* FC_LONG */	0x12, 0x10, /*
/* 822 */	0x8, /*	FC_UP [pointer_deref] */
0x1a, /*	FC_LONG */	/* 914 */ NdrFcShort(0xfda4), /* Offset= -604 (310) */
FC_BOGUS_STRUCT */	/* 874 */ 0x40, /* FC_STRUCTPAD4 */	/* 916 */
0x3, /* 3 */	0x4c, /*	0x12, 0x10, /*
/* 824 */ NdrFcShort(0x10), /* 16 */	FC_EMBEDDED_COMPLEX */	FC_UP [pointer_deref] */
/* 826 */ NdrFcShort(0x0), /* 0 */	/* 876 */ 0x0, /* 0 */	/* 918 */ NdrFcShort(0xfdba), /* Offset= -582 (336) */
/* 828 */ NdrFcShort(0x6), /* Offset= 6 (834) */	NdrFcShort(0xfe0f), /*	/* 920 */
/* 830 */ 0x8, /* FC_LONG */	Offset= -497 (380) */	0x12, 0x10, /*
0x40, /*	0x5b, /*	FC_UP [pointer_deref] */
FC_STRUCTPAD4 */	FC_END */	/* 922 */ NdrFcShort(0xfdc8), /* Offset= -568 (354) */
/* 832 */ 0x36, /* FC_POINTER */	/* 880 */	/* 924 */
0x5b, /*	0x12, 0x0, /* FC_UP */	0x12, 0x10, /*
FC_END */	/* 882 */ NdrFcShort(0xff04), /* Offset= -252 (630) */	FC_UP [pointer_deref] */
/* 834 */	/* 884 */	/* 926 */ NdrFcShort(0xfdd6), /* Offset= -554 (372) */
0x12, 0x0, /* FC_UP */	0x12, 0x8, /* FC_UP	/* 928 */
/* 836 */ NdrFcShort(0xffe6), /* Offset= -26 (810) */	[simple_pointer] */	0x12, 0x10, /*
/* 838 */	/* 886 */ 0x1, /* FC_BYTE */	FC_UP [pointer_deref] */
0x15, /*	0x5c, /*	/* 930 */ NdrFcShort(0x2), /* Offset= 2 (932) */
FC_STRUCT */	FC_PAD */	/* 932 */
0x3, /* 3 */	/* 888 */	0x12, 0x0, /* FC_UP */
/* 840 */ NdrFcShort(0x8), /* 8 */	0x12, 0x8, /* FC_UP	/* 934 */ NdrFcShort(0x14), /* Offset= 20 (954) */
/* 842 */ 0x8, /* FC_LONG */	[simple_pointer] */	/* 936 */
0x8, /*	/* 890 */ 0x6, /* FC_SHORT */	0x15, /*
FC_LONG */	0x5c, /*	FC_STRUCT */
/* 844 */ 0x5c, /* FC_PAD */	FC_PAD */	0x7, /* 7 */
0x5b, /*	/* 892 */	/* 938 */ NdrFcShort(0x10), /* 16 */
FC_END */	0x12, 0x8, /* FC_UP	/* 940 */ 0x6, /* FC_SHORT */
/* 846 */	[simple_pointer] */	0x1, /*
0x1b, /*	/* 894 */ 0x8, /* FC_LONG */	FC_BYTE */
FC_CARRAY */	0x5c, /*	/* 942 */ 0x1, /* FC_BYTE */
0x3, /* 3 */	FC_PAD */	0x8, /*
/* 848 */ NdrFcShort(0x8), /* 8 */	/* 896 */	FC_LONG */
/* 850 */ 0x7, /* Corr desc: FC_USHORT */	0x12, 0x8, /* FC_UP	/* 944 */ 0xb, /* FC_HYPER */
0x0, /* */	[simple_pointer] */	0x5b, /*
/* 852 */ NdrFcShort(0xffc8), /* -56 */	/* 898 */ 0xb, /* FC_HYPER */	FC_END */
/* 854 */ NdrFcShort(0x1), /* Corr flags: early, */	0x5c, /*	/* 946 */
/* 856 */ 0x4c, /*	FC_PAD */	0x12, 0x0, /* FC_UP */
FC_EMBEDDED_COMPLEX */	/* 900 */	/* 948 */ NdrFcShort(0xfff4), /* Offset= -12 (936) */
0x0, /* 0 */	0x12, 0x8, /* FC_UP	/* 950 */

```

0x12, 0x8, /* FC_UP
[simple_pointer] */
/* 952 */ 0x2, /* FC_CHAR */
0x5c, /*
FC_PAD */
/* 954 */
0x1a, /*
FC_BOGUS_STRUCT */
0x7, /* 7 */
/* 956 */ NdrFcShort( 0x20 ), /* 32 */
/* 958 */ NdrFcShort( 0x0 ), /* 0 */
/* 960 */ NdrFcShort( 0x0 ), /* Offset= 0 (960) */
/* 962 */ 0x8, /* FC_LONG */
0x8, /*
FC_LONG */
/* 964 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 966 */ 0x6, /* FC_SHORT */
0x6, /*
FC_SHORT */
/* 968 */ 0x4c, /*
FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 970 */ NdrFcShort( 0xfc3c ), /* Offset= -964 (6) */
/* 972 */ 0x5c, /* FC_PAD */
0x5b, /*
FC_END */
/* 974 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /* 131 */
/* 976 */ NdrFcShort( 0x0 ), /* 0 */
/* 978 */ NdrFcShort( 0x18 ), /* 24 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0xfc2c ), /* Offset= -980 (2) */
/* 984 */
0x11, 0x4, /* FC_RP
[allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /* Offset= 6 (992) */
/* 988 */
0x13, 0x0, /* FC_OP */
/* 990 */ NdrFcShort( 0xffdc ), /* Offset= -36 (954) */
/* 992 */ 0xb4, /* FC_USER_MARSHAL */
0x83, /* 131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xff4 ), /* Offset= -12 (988) */
0x0
}
};

```

```

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =

```

```

{
{
VARIANT_UserSize
,VARIANT_UserMarshal
,VARIANT_UserUnmarshal
,VARIANT_UserFree
}
};
/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */
/* Object interface: IUnknown, ver. 0.0,
GUID={0x00000000,0x0000,0x0000,{0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */
/* Object interface: ITPCC, ver. 0.0,
GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */
#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
0,
44,
88,
132,
176,
220
};
static const MIDL_STUBLESS_PROXY_INFO
ITPCC_ProxyInfo =
{
&Object_StubDesc,
__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0
};

```

```

static const MIDL_SERVER_INFO ITPCC_ServerInfo =

```

```

{
&Object_StubDesc,
0,
__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0,
0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
&ITPCC_ProxyInfo,
&IID_ITPCC,
IUnknown_QueryInterface_Proxy,
IUnknown_AddRef_Proxy,
IUnknown_Release_Proxy,
(void *) (INT_PTR) -1 /* ITPCC::NewOrder */,
(void *) (INT_PTR) -1 /* ITPCC::Payment */,
(void *) (INT_PTR) -1 /* ITPCC::Delivery */,
(void *) (INT_PTR) -1 /* ITPCC::StockLevel */,
(void *) (INT_PTR) -1 /* ITPCC::OrderStatus */,
(void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};
const CInterfaceStubVtbl _ITPCCStubVtbl =
{
&IID_ITPCC,
&ITPCC_ServerInfo,
9,
0, /* pure interpreted */
CStdStubBuffer_METHODS
};
static const MIDL_STUB_DESC Object_StubDesc =
{
0,
NdrOleAllocate,
NdrOleFree,
0,
0,
0,
0,
0,
0,
__MIDL_TypeFormatString.Format,
1, /* -error bounds_check flag */
0x50002, /* Ndr library version */
0,
0x6000169, /* MIDL Version 6.0.361 */
0,
UserMarshalRoutines,
0, /* notify & notify_flag routine table */
0x1, /* MIDL flag */
0, /* cs routines */
0, /* proxy/server info */

```

```

0 /* Reserved5 */
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    (CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    (CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n)
IID_GENERIC_CHECK_IID(_tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int *
pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) & _tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) & _tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &
    _tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};
#endif _MSC_VER >= 1200

```

```

#pragma warning(pop)
#endif

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

tpcc_com_sl.rgs

HKCR
{
    TPCC.StockLevel.1 = s 'StockLevel Class'
    {
        CLSID = s '{2668369E-A50D-11D2-
BA4E-00C04FBFE08B}'
    }
    TPCC.StockLevel = s 'StockLevel Class'
    {
        CurVer = s 'TPCC.StockLevel.1'
    }
    NoRemove CLSID
    {
        ForceRemove {2668369E-A50D-11D2-
BA4E-00C04FBFE08B} = s 'StockLevel Class'
        {
            ProgID = s
'TPCC.StockLevel.1'
            VersionIndependentProgID = s
'TPCC.StockLevel'
            InprocServer32 = s
'%MODULE%'
            {
                val ThreadingModel
                = s 'Both'
            }
        }
    }
}

```

tpcc_odbc.cpp

```

/* FILE: TPCC_ODBC.CPP
* Microsoft TPC-C
Kit Ver. 4.42.000
* Copyright Microsoft,
2002
* All Rights Reserved
*
* Version 4.10.000
audited by Richard Gimarc, Performance Metrics, 3/17/99
*
* PURPOSE: Implements ODBC calls for

```

```

TPC-C txns.
* Contact: Charles Levine (clevine@microsoft.com)
*
* Change history:
* 4.42.000 - changed w_id fields from short
to long to support >32K warehouses
* 4.20.000 - updated rev number to match
kit
* 4.10.001 - not deleting error class in catch
handler on deadlock retry;
* not a functional
bug, but a memory leak
*/

#include <windows.h>
#include <stdio.h>
#include <assert.h>

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>

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

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

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

            DisableThreadLibraryCalls(hModule);
            if
(SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                return FALSE;
            break;

        case DLL_PROCESS_DETACH:
            if (henv != NULL)
                SQLFreeEnv(henv);
            break;

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

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
*
*/

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

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

```

```

{ 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_erno == errorMsgs[i].iError )
        break;
}
if ( !errorMsgs[i].szMsg[0] )
    return szNotFound;
else
    return errorMsgs[i].szMsg;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
LPCSTR szServer, // name of SQL
server
LPCSTR szUser, // user
name for login
LPCSTR szPassword, // password for
login
LPCSTR szHost, // not
used
LPCSTR szDatabase, // name of database
to use
LPCWSTR szSPPrefix, // prefix to append
to the stored procedure names
BOOL bCallNoDuplicatesNewOrder // whether to
check for non-duplicate items in NewOrder and call a new SP
{
    return new CTPCC_ODBC( szServer, szUser,
szPassword, szHost, szDatabase, szSPPrefix,
bCallNoDuplicatesNewOrder );
}

CTPCC_ODBC::CTPCC_ODBC (
LPCSTR szServer,
// name of SQL server
LPCSTR szUser,
// user name for login
LPCSTR szPassword,
// password for login
LPCSTR szHost,

```

```

// not used
LPCSTR szDatabase,
// name of database to use
LPCWSTR szSPPrefix,
// prefix to append to the stored procedure
names
BOOL bCallNoDuplicatesNewOrder // whether to
check for non-duplicate items in NewOrder and call a new SP
)
: m_bCallNoDuplicatesNewOrder(bCallNoDuplicatesNewOrder)
{
    RETCODE rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

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

    m_descNewOrderCols1 = SQL_NULL_HDESC;
    m_descNewOrderCols2 = SQL_NULL_HDESC;
    m_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_S
P_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;
    SDWORD            INativeError;
    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, &INativeError,
        (BYTE
*)&szMsg, sizeof(szMsg), NULL);
            if (rc == SQL_NO_DATA)
                {
                    break;
                }

            if (rc != SQL_SUCCESS)
                {
                    break;
                }

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

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

            // quit if there isn't enough room to
concatenate error text
            if ( ( strlen(szMsg) + 2) > (sizeof(szTmp) -
strlen(szTmp)) )
                break;

            // include line break after first error msg
            if (szTmp[0] != 0)
                strcat( szTmp, "\n");
            strcat( szTmp, szMsg );
        }

        if (pODBCErr->m_odbcerrstr != NULL)
        {
            delete [] pODBCErr->m_odbcerrstr;
            pODBCErr->m_odbcerrstr = NULL;
        }

```

```

        if (strlen(szTmp) > 0)
        {
            pODBCErr->m_odbcerrstr = new
char[ strlen(szTmp)+1 ];
            strcpy( pODBCErr->m_odbcerrstr,
szTmp );
        }

        SQLFreeStmt(m_hstmt, SQL_CLOSE);
        throw pODBCErr;
    }

void CTPCC_ODBC::InitStockLevelParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtStockLevel) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtStockLevel;

    int i = 0;
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.StockLevel.w_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0,
0, &m_txn.StockLevel.d_id, 0, NULL) != SQL_SUCCESS
        || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SSHORT, SQL_SMALLINT, 0,
0, &m_txn.StockLevel.threshold, 0, NULL) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eBindParam);

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

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

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

    m_hstmt = m_hstmtStockLevel;

    while (TRUE)

```

```

    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt, m_szStockLevelCommand,
SQL_NTS);
            if (rc != SQL_SUCCESS &&
rc != SQL_SUCCESS_WITH_INFO)
                ThrowError(CODBCERR::eExecDirect);

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

            SQLFreeStmt(m_hstmt,
SQL_CLOSE);

            m_txn.StockLevel.exec_status_code = eOK;
            break;
        }
        catch (CODBCERR *e)
        {
            if (!(e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))
                throw;

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

        // if (iTryCount)
        // throw new
        CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_
TRANS, iTryCount);
    }

void CTPCC_ODBC::InitNewOrderParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtNewOrder) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrderNoDuplicates) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols1) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;

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

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

    for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS;
j++)
    {
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER,
0, 0, &m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
            || SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_quantity, 0,
NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);
    }
}

```

```

&m_descNewOrderCols2) != SQL_SUCCESS
    ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols1) != SQL_SUCCESS
    ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;

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

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

    for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS;
j++)
    {
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER,
0, 0, &m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
            || SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_quantity, 0,
NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);
    }
}

```

```

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

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

// associate the column bindings for the second result
set
if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descNewOrderCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.w_tax, 0, NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.NewOrder.o_id, 0, NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_last,
sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.c_discount, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_credit,
sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,

```

```

SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

//Compose the New Order statement
_snwprintf(m_szNewOrderCommand,
sizeof(m_szNewOrderCommand)/sizeof(m_szNewOrderComma
nd[0]),
    // 0 1 2
    // 012345678901234567890123456789
    L"{call
%stpcpc_neworder(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)"
    L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?}";
m_szSPPrefix);

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

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

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

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

```

```

SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0,
0, &m_txn.NewOrder.o_all_local, 0, NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindParam);

for (int j=0; j<MAX_OL_NEW_ORDER_ITEMS;
j++)
{
    if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.NewOrder.OL[j].ol_i_id, 0, NULL) != SQL_SUCCESS
        // SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER,
0, 0, &m_txn.NewOrder.OL[j].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
        // SQLBindParameter(m_hstmt,
++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.NewOrder.OL[j].ol_quantity, 0,
NULL) != SQL_SUCCESS
        )
        ThrowError(CODBCERR::eBindParam);
}

// set row-wise binding
if ( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.NewOrder.OL[0]),
SQL_IS_UIINTEGER) != SQL_SUCCESS
    // SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name,
sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.NewOrder.OL[0].ol_stock, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) !=
SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_i_price, 0,
NULL) != SQL_SUCCESS
    // SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

```

```

// associate the column bindings for the second result
set
if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC,
m_descNewOrderNoDuplicatesCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if ( SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.w_tax, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.d_tax, 0, NULL) !=
SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.NewOrder.o_id, 0, NULL) !=
SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_last,
sizeof(m_txn.NewOrder.c_last), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.NewOrder.c_discount, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.NewOrder.c_credit,
sizeof(m_txn.NewOrder.c_credit), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.NewOrder.o_entry_d, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_no_commit_flag, 0, NULL) !=
SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindCol);

//Compose the New Order statement
_snwprintf(m_szNewOrderNoDuplicatesCommand,
sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNew
OrderNoDuplicatesCommand[0]),
L"{call
%stpc_neworder_new(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
,?,"
m_szSPPrefix);

m_iBeginNewOrderNoDuplicatesVariablePart = 33 +
wcslen(m_szSPPrefix); // fixed part + prefix part
}

//
// Returns true if there are duplicate (warehouse_id,
item_id)

```

```

// lineitem pairs in New Order input parameters.
//
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
    int i, j;

    for (i = 0; i < m_txn.NewOrder.o_ol_cnt; ++i)
    {
        for (j = i+1; j < m_txn.NewOrder.o_ol_cnt;
++j)
        {
            if
(m_txn.NewOrder.OL[i].ol_i_id ==
m_txn.NewOrder.OL[j].ol_i_id)
            {
                return true;
            }
        }
    }

    return false;
}

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

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

```

1 2

// 0

```

012345678901234567890123456789 //
wchar_t
szSqlTemplate[iMAX_SP_NAME_LEN];

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

m_hstmt = m_hstmtNewOrder;

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

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

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

while (TRUE)
{
    try
    {

```

```

        m_BindOffset = 0;
        rc =
SQLExecDirectW(m_hstmt, szSqlTemplate, SQL_NTS);
        if (rc != SQL_SUCCESS &&
rc != SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::eExecDirect);

        // Get order line results
        m_txn.NewOrder.total_amount
= 0;
        for (i = 0;
i < m_txn.NewOrder.o_ol_cnt; i++)
        {
            // set the bind offset
            value...
            m_BindOffset = i *
sizeof(m_txn.NewOrder.OL[0]);

            if
(SQLFetch(m_hstmt) == SQL_ERROR)

                ThrowError(CODBCERR::eFetch);

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

                ThrowError(CODBCERR::eMoreResults);

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

        // associate the column
bindings for the second result set
        if
(SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descNewOrderCols2, SQL_IS_POINTER ) !=
SQL_SUCCESS )

            ThrowError(CODBCERR::eSetStmtAttr);

        if ( SQLFetch(m_hstmt) ==
SQL_ERROR)

            ThrowError(CODBCERR::eFetch);

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        if (m_no_commit_flag == 1)

```

```

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

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else
            m_txn.NewOrder.exec_status_code = eInvalidItem;

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

                throw;

            // hit deadlock; backoff for
            increasingly longer period

            delete e;
            Sleep(10 * iTryCount);
        }
    }

    // if (iTryCount)
    // throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_
TRANS, iTryCount);
}

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

    1    2    3

    0123456789012345678901234567890123

    wchar_t
    szSqlTemplate[iMAX_SP_NAME_LEN];

    // L" {call

```

```

tpcc_neworder_new(?,?,?,?,?"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?";

        m_hstmt = m_hstmtNewOrderNoDuplicates;

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

            ThrowError(CODBCERR::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);

        if (m_no_commit_flag == 1)
        {

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

            m_txn.NewOrder.exec_status_code = eOK;
        }
        else

            m_txn.NewOrder.exec_status_code = eInvalidItem;

            break;
        }
        catch (CODBCERR *e)
        {

            if (!(e->m_bDeadLock) ||
(++iTryCount > iMaxRetries))

                throw;

                // hit deadlock; backoff for
increasingly longer period

                delete e;
                Sleep(10 * iTryCount);
            }
        }

        //
        if (iTryCount)
            throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_
TRANS, iTryCount);
    }

    void CTPCC_ODBC::InitPaymentParams()
    {

        if ( SQLAllocHandle(SQL_HANDLE_STMT,

```

```

m_hdbc, &m_hstmtPayment) != SQL_SUCCESS )
        ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtPayment;

        int i = 0;
        if ( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_DOUBLE, SQL_NUMERIC, 6,
2, &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0,
0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0,
0, &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
            // SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);

            i = 0;
            if ( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.Payment.c_id,
0, NULL) !=
SQL_SUCCESS
                // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) !=
SQL_SUCCESS
                // SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.Payment.h_date,
0, NULL) != SQL_SUCCESS
                // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_1,
sizeof(m_txn.Payment.w_street_1), NULL) !=
SQL_SUCCESS
                // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_street_2,
sizeof(m_txn.Payment.w_street_2), NULL) !=
SQL_SUCCESS
                // SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.Payment.w_city,
sizeof(m_txn.Payment.w_city), NULL) !=
SQL_SUCCESS

```

```

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

```

```

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

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

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

    m_hstmt = m_hstmtPayment;

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

    while (TRUE)
    {
        try
        {
            rc =
SQLExecDirectW(m_hstmt, m_szPaymentCommand,
SQL_NTS);

```

```

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

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

        SQLFreeStmt(m_hstmt,
SQL_CLOSE);

        if (m_txn.Payment.c_id == 0)
            throw new
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_INVALID_
CUST);
        else
            m_txn.Payment.exec_status_code = eOK;

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

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

    // if (iTryCount)
    // throw new
CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_
TRANS, iTryCount);
}

void CTPCC_ODBC::InitOrderStatusParams()
{
    if ( SQLAllocHandle(SQL_HANDLE_STMT,
m_hdbc, &m_hstmtOrderStatus) != SQL_SUCCESS
        ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
        ||
SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);
}

```

```

m_hstmt = m_hstmtOrderStatus;

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

int i = 0;
if( SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.w_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_UTINYINT, SQL_TINYINT, 0,
0, &m_txn.OrderStatus.d_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_SLONG, SQL_INTEGER, 0, 0,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
    || SQLBindParameter(m_hstmt, ++i,
SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR,
sizeof(m_txn.OrderStatus.c_last), 0, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindParam);

// configure block cursor
if( SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) !=
SQL_SUCCESS
    || SQLSetStmtAttrW(m_hstmt,
SQL_ATTR_ROWS_FETCHED_PTR, &m_RowsFetched, 0) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) !=
SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.OL[0].ol_i_id, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.OL[0].ol_quantity, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.OL[0].ol_amount, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) !=
SQL_SUCCESS
    )

```

```

ThrowError(CODBCERR::eBindCol);

if( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2,
SQL_IS_POINTER ) != SQL_SUCCESS )
    ThrowError(CODBCERR::eSetStmtAttr);

i = 0;
if( SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.c_id, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_last,
sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_first,
sizeof(m_txn.OrderStatus.c_first), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_CHAR, &m_txn.OrderStatus.c_middle,
sizeof(m_txn.OrderStatus.c_middle), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_TYPE_TIMESTAMP, &m_txn.OrderStatus.o_entry_d,
0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SSHORT, &m_txn.OrderStatus.o_carrier_id, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_DOUBLE, &m_txn.OrderStatus.c_balance, 0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i,
SQL_C_SLONG, &m_txn.OrderStatus.o_id, 0, NULL) !=
SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

//Compose Order Status statement
_snwprintf(m_szOrderStatusCommand,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOrderStatusCom
mand[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
    _snwprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryCommand[
0]),
        L" {call %stpc_delivery (?,?)",
m_szSPPrefix);
}

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

    m_hstmt = m_hstmtDelivery;

    while (TRUE)
    {
        try
        {

```

```

rc =
SQLExecDirectW(m_hstmt, m_szDeliveryCommand,
SQL_NTS);
        if (rc != SQL_SUCCESS &&
rc != SQL_SUCCESS_WITH_INFO)

            ThrowError(CODBCERR::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:          TPCC_ODBC.H
*
*          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_erno = iErr; m_iTryCount = 0; };

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

int m_erno;
int m_iTryCount;

int ErrorType() {return
ERR_TYPE_TPCC_ODBC;};
char* ErrorTypeStr() { return "TPCC
ODBC"; }
int ErrorNum() {return
m_erno;};
char* ErrorText();
};

class 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_LEN];
    int
    m_iBeginNewOrderVariablePart;// begining of the
variable part in NewOrder statement
    int
    m_iBeginNewOrderNoDuplicatesVariablePart;
// begining of the variable part in NewOrder statement
    wchar_t
    m_szPaymentCommand[iMAX_SP_NAME_LEN];
    wchar_t
    m_szDeliveryCommand[iMAX_SP_NAME_LEN];
    wchar_t
    m_szOrderStatusCommand[iMAX_SP_NAME_LEN
];
    wchar_t
    m_szStockLevelCommand[iMAX_SP_NAME_LEN];

// new-order specific fields
SQLINTEGER          m_BindOffset;
SQLINTEGER          m_RowsFetched;
    int
    m_no_commit_flag;

// tpcc_neworder_new flag
    BOOL
    m_bCallNoDuplicatesNewOrder;

//void
ThrowError( CODBCERR::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( LPCSTR szServer,
LPCSTR szUser, LPCSTR szPassword,
LPCSTR
szHost, LPCSTR szDatabase,
LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder);
        ~CTPCC_ODBC(void);

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

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

};

// wrapper routine for class constructor
extern "C" DIIDecl CTPCC_ODBC* CTPCC_ODBC_new

```

```

( LPCSTR szServer, LPCSTR szUser,
LPCSTR szPassword,
LPCSTR szHost, LPCSTR szDatabase,
LPCWSTR szSPPrefix, BOOL
bCallNoDuplicatesNewOrder );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR,
LPCSTR, LPCSTR, LPCSTR, LPCWSTR, BOOL);

```

trans.h

```

/* FILE: TRANS.H
* 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: 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 dblink, 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.
#ifdef __SQLTYPES
    typedef struct
    {
        short
        /* SQLSMALLINT */ year;
        SQLSMALLINT /* month;
        unsigned short /*
        SQLSMALLINT /* day;
        unsigned short /*
        SQLSMALLINT /* hour;
        unsigned short /*
        SQLSMALLINT /* minute;
        unsigned short /*
        SQLSMALLINT /* second;
        unsigned long /*
        SQLINTEGER /* fraction;
    } TIMESTAMP_STRUCT;
#else
// 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
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     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.
#ifndef DllDecl
#define DllDecl __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
public:
    CTPCC_BASE(void) {};
    virtual ~CTPCC_BASE(void) {};

    virtual PNEW_ORDER_DATA
    BuffAddr_NewOrder() = 0;
    virtual PPAYMENT_DATA
    BuffAddr_Payment() = 0;
    virtual PDELIVERY_DATA
    BuffAddr_Delivery() = 0;
    virtual PSTOCK_LEVEL_DATA
    BuffAddr_StockLevel() = 0;

```

```

    virtual PORDER_STATUS_DATA
    BuffAddr_OrderStatus() = 0;

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

```

txnlog.h

```

/*      FILE:          TXNLOG.H
*      Microsoft TPC-C
Kit Ver. 4.10.000
*      not yet audited
*      PURPOSE:      Header file for txn log class
*      Copyright Microsoft,
1999
*      All Rights Reserved
*
*/
#include <stdio.h>    //needed for FILE

#define DRIVER_NAME_LEN
32    //max length of the driver
engine name - must be the same as in engstut.h!
#define TXN_LOG_INCORRECTLY_SHUT_DOWN 100
//ctrl rec subtype generated by the txn log when
reading an abruptly shut down log

#pragma once

typedef struct _TXN_NEWORDER
{
    BYTE      OL_Count;    //range 0 to 31
    BYTE      OL_Remote_Count;    //range 0 to 31
    WORD      c_id;
    int       o_id;
} TXN_NEWORDER;

typedef struct _TXN_PAYMENT
{
    BYTE      CustByName;
    BYTE      IsRemote;
} TXN_PAYMENT;

typedef struct _TXN_ORDERSTATUS

```

```

{
    BYTE    CustByName;
} TXN_ORDERSTATUS;

typedef union _TXN_DETAILS
{
    TXN_NEWORDER  NewOrder;
    TXN_PAYMENT   Payment;
    TXN_ORDERSTATUS
OrderStatus;
} TXN_DETAILS;

// Common header for all records in txn log. The
TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL
1 //
#define TXN_REC_TYPE_TPCC
2 // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF
3
#define TXN_REC_TYPE_TPCW
4 // replaces
TRANSACTION_TYPE_TPCW

typedef struct _TXN_RECORD_HEADER
{
    JULIAN_TIME    TxnStartT0;
    // start of txn
    BYTE    TxnType;
    // one of TXN_REC_TYPE_*
    BYTE    TxnSubType;
    // depends on TxnType
} TXN_RECORD_HEADER;
*PTXN_RECORD_HEADER;

typedef struct _TXN_RECORD_CONTROL
{
    // common header; must exactly match
TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;
    // start of txn
    BYTE    TxnType;
    // = TXN_REC_TYPE_CONTROL
    BYTE    TxnSubType;
    // depends on TxnType
    // end of common header

    DWORD    Len;
    // number of bytes after this field
} TXN_RECORD_CONTROL;

```

```

*PTXN_RECORD_CONTROL;

// TPC-C Txn Record Layout:
//
// 'TxnStartT0' is a Julian timestamp corresponding to the
moment the
// txn is sent to the SUT, i.e., beginning of response time.
Deltas
// are in milliseconds. Note that if RTDelay > 0, then the txn
was
// delayed by this amount. The delay occurs at the beginning of
the
// response time. So if RTDelay > 0, then the txn was actually
sent
// at TxnStartT0 + RTDelay.
//
// Graphically:
//
// time -->
//
// |--- Menu ---|--- Keying ---|--- Response ---|--- Think ---|
// <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4 -> <- DeltaT3 ->
//
// ^
// ^ TxnStartT0
//
// RTDelay is the amount of response time delay included in
DeltaT4.
// RTDelay is recorded per txn because this value can be
changed on
// the fly, and so may vary from txn to txn.
//
// TxnStatus is the txn completion code. It is used to indicate
errors.
// For example, in the New Order txn, 1% of txns abort.
TxnStatus will
// reflect this.

typedef struct _TXN_RECORD_TPCC
{
    // common header; must exactly match
TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;
    // start of txn
    BYTE    TxnType;
    // = TXN_REC_TYPE_TPCC
    BYTE    TxnSubType;
    // depends on TxnType
    // end of common header

    int    DeltaT1; // menu
    time (ms)
    int    DeltaT2; // keying
    time (ms)

```

```

    int    DeltaT3; // think
    time (ms)
    int    DeltaT4; //
    response time (ms)
    int    RTDelay; //
    response time delay (ms)
    int    TxnError;
    // error code providing more detail for TxnStatus
    int    w_id;
    // warehouse ID
    BYTE    d_id;
    // assigned district ID for this thread
    BYTE    d_id_ThisTxn; // district
ID chosen for this particular
    BYTE    TxnStatus; //
completion status for txn to indicate errors
    BYTE    reserved; // for
word alignment
    TXN_DETAILS    TxnDetails;
    //

    bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
} TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

// TPC-C Deferred Delivery Txn Record Layout:
//
// Incorporating delivery transaction information into the above
// structure would increase the size of TXN_DETAILS
from 8 to 42 bytes.
// Hence, we store delivery transaction details in a separate
structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
    // common header; must exactly match
TXN_RECORD_HEADER
    JULIAN_TIME    TxnStartT0;
    // start of txn
    BYTE    TxnType;
    // = TXN_REC_TYPE_TPCC_DELIV_DEF
    BYTE    TxnSubType;
    // = 0
    // end of common header

    int    DeltaT4; //
    response time (ms)
    int    DeltaTxnExec;
    // execution time (ms)
    int    w_id;
    // warehouse ID
    BYTE    TxnStatus; //

```

```

completion status for txn to indicate errors
        BYTE reserved; // for
word alignment
        short o_carrier_id; // carrier id
        long o_id[10]; //
returned delivery transaction ids

        bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus ==
ERR_TYPE_DELIVERY_POST); }
        } TXN_RECORD_TPCC_DELIV_DEF,
*PTXN_RECORD_TPCC_DELIV_DEF;

//
//TPC-W records.
//
typedef struct _TXN_RECORD_TPCW
{
// common header; must exactly match
TXN_RECORD_HEADER
        JULIAN_TIME TxnStartT0;
// start of txn
        BYTE TxnType;
// = TXN_REC_TYPE_TPCW
        BYTE TxnSubType;
// depends on TxnType
// end of common header

        int ThinkTime; // think
time (ms)
        int WIRT;
// response time (ms)
        int TxnError;
// error code providing more detail for TxnStatus
        BYTE TxnStatus; //
completion status for txn to indicate errors
//This field below depends on the txn sub
type:
        //- for Home interaction: it
indicates whether the user was a new customer (or returning)
        //- for Buy Confirm: it
indicates whether the shipping address was updated
        //- for Search Request:
it indicates the search type (Author, Title, or Subject)
//This statistics needs to be reported
according to 5.5.5.1 clause in the specs.
//Because this field occupies 1 byte, the
record structure is already aligned on word boundary.
        union {
                BYTE newCustomer;
                BYTE addrUpdated;
                BYTE searchType;
        } intrDetails;

```

```

//This field is mostly for
informational/debugging purposes.
//It indicates what user performed this web
interaction and what instance (session) of that use it was.
//The first 22 bits indicate the user #, and
the top 10 bits indicate instance (session) #.
        unsigned __int32 uiUser;

        bool IsSuccessRecord() { return
(TxnStatus == ERR_SUCCESS); }
        } TXN_RECORD_TPCW,
*PTXN_RECORD_TPCW;

//
// Data part of a control record written when
a user is created (or it's new session) - to record USMD
typedef struct
_TXN_RECORD_TPCW_USER_DATA
{
        unsigned __int32 uiUser;
// user number
        JULIAN_TIME
        USMD; // USMD
for this user
        BYTE
        bRetCust; // returning
customer?
        } TXN_RECORD_TPCW_USER_DATA,
*PTXN_RECORD_TPCW_USER_DATA;

//The entire TPCW User control record structure
typedef struct _TXN_RECORD_TPCW_USER
{
// common header; must exactly match
TXN_RECORD_HEADER
        JULIAN_TIME TxnStartT0;
// start of txn
        BYTE TxnType;
// = TXN_REC_TYPE_CONTROL
        BYTE TxnSubType;
// depends on TxnType
// end of common header

        DWORD Len;
// number of bytes after this field
//The fields above must exactly match
TXN_RECORD_CONTROL

//The fields below must exactly match
TXN_RECORD_TPCW_USER_DATA
        unsigned __int32 uiUser;
// user number
        JULIAN_TIME

```

```

        USMD; // USMD
for this user
        BYTE
        bRetCust; // returning
customer?
        } TXN_RECORD_TPCW_USER,
*PTXN_RECORD_TPCW_USER;

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

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

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

```

```

        // driver engine that created this log file
        char
        szDriverEngineName[DRIVER_NAME_LEN];
        // the record map provides a fast way to
get close to a particular timestamp in a sorted log file.
//
//      struct
//      {
//          JULIAN_TIME
record      TS;          // timestamp of
//
//          int
file        iPos;      // byte position in
//
//      }
//      RecMap[RecMapSize];
#define     RecMapSize
200
    } TXN_LOG_HEADER, *PTXN_LOG_HEADER;

/* Header of the sorted pointers blocks in Temp file
(in merging). */
typedef struct BLOCK_HEADER {
    long      BlockPos;
    _int64    CurPos;
    DWORD     BytesRead;
    int       nRecords;
    BYTE      *offset; /* offset of pointers
to records in the log file */
} BLOCK_HEADER, *PBLOCK_HEADER;

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

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

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

#define     TXN_LOG_OS_ERROR      1
#define     TXN_LOG_NOT_SORTED   2

```

```

#define SKIP_CTRL_RECS      1

class CTxnLog
{
private:
    DWORD      iBufferSize;
//buffer allocated size
    DWORD      iBytesFreeInBuffer;
//total bytes available for use in buffer
    int
iNumBuffers;          //buffers
in use
    int
iActiveBuffer;
//indicates which buffer is active: 0 or 1
    int        iIoBuffer;
//buffer for any
pending IO operation
//      int
iFilePointer;
//position in file.
    LARGE_INTEGER iFilePointer;
//position in file.
    int
iNextRec;          //when
reading, ordinal value of next record

// A "save point" is remembered each time
getNextRecord is called with a start time specified.
// The next time it is called, if start time is
after the save point, we start scanning from the
// save point. This is particularly useful in
FindBestInterval, where the log is scanned repeatedly.
    JULIAN_TIME
SavePtTime;
//
//      int
iSavePtFilePointer;
    LARGE_INTEGER iSavePtFilePointer;
    int
iSavePtNextRec;

    JULIAN_TIME lastTS;
//when writing
sorted output, used to verify records are sorted
    BOOL bWrite;
//writing log file
    BOOL bCrashOpen;
// tolerate bad
headers and consistency checks
    BOOL bLogSorted;
// is log
file sorted? applies to both input and output
    JULIAN_TIME

```

```

BeginTxnTS;          //
timestamp of first (lowest) txn start
    JULIAN_TIME
EndTxnTS;          //
timestamp of last (highest) txn completion time
    int
iRecCount;          //
number of records in log file
// To write a checkpoint information into
the header, need to know the EndTxnTS for the
// last record written to the disk. It is not
necessarily the last record in the
// last written buffer, as the last record may
be only partially in the buffer.
// So remember the timestamps for 2 last
records that begin in the buffer - one of
// them will be the last complete record
written to disk.
    JULIAN_TIME
PrevEndTxnTS;      // timestamp of the
previous to last record
    union {
        TXN_LOG_HEADER
HeaderForCheckpoint; // header written on
every checkpoint
        char
szHeaderBuffer[512]; // 512 bytes is the
minimum we can write to the disk
    } HeaderBuffer; //need the union
because can't write sizeof(TXN_LOG_HEADER) - too few
bytes
// Control record returned from
getNextRecord if the file
// currently opened for read was not
properly shut down
    struct
    {
        TXN_RECORD_CONTROL
RecHeader;
        char
szDriverName[DRIVER_NAME_LEN];
    } IncorrectShutDownRec;
    BYTE *pCurrent;
//ptr to current buffer
    BYTE
*pBuffer[MAX_NUM_BUFFERS];
    PTXN_RECORD_HEADER
*TxnArray;          //transaction record
pointer array for sort

```



```

        DWORD          dwError;
        DWORD          dwCheckpointError;
        //error in checkpoint thread
        HANDLE         hTxnFile;
        //handle to log file
        HANDLE         hMapFile;
        //map file used when sorting

the log
        HANDLE         hIoComplete;
        //event to signify that there are
no pending IOs
        HANDLE         hLogFileIo;
        //event to signal the
IO thread to write the inactive buffer
        HANDLE
        hStopCheckpointThread; //event to signal the
checkpoint thread to exit

        Spinlock Spin;
        //spin lock to protect the txn
log file buffers
        Spinlock WriteSpin;
        //spin lock to protect the WriteFile
operation between IO and Checkpoint threads

        FILE
        *tmpFile; //temp file for merging sorted
pieces
        PBLOCK_HEADER
        tmpHeaders; //sorted pointers
block header
        BYTE
        **recPointers; //record pointer buffers for
each sorted block
        PTXN_RECORD_HEADER
        *recBuffers; //record buffers for each sorted
block
        int          *PointersRead;
        //# of pointers
processed in each block
        BOOL          *BlockAvailable;
        //whether to check a particular block for
jmin
        int          nBlocks;
        int          jmin;
        //index
(block-wise) of the lowest timestamp record
        int          iAvgRecordLen;
        //average record
length
        int

```

```

        iSortedReturnedCount; //keeps
track of the # of sorted records returned through
GetSortedRecord()

        BOOL          bIncorrectShutDown;
        // indicates whether the log opened for
read was not correctly shut down

        int Write(BYTE *ptr, DWORD Size);
        static void LogFileIO(CTxnLog *);

        void LoadBuffers(int j);
        //used in sort/merge to load record buffers

        static void CheckpointThread(CTxnLog *);
        // checkpointing thread

public:
        CTxnLog(LPCTSTR szFileName,
        DWORD dwOpts, char *szDriver = NULL);
        ~CTxnLog(void);

        int WriteToLog(PTXN_RECORD_TPCC
        pTxnRcrd);
        int
        WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
        int
        WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
        int
        WriteToLog(PTXN_RECORD_HEADER pCtrlRec);
        int WriteToLog(PTXN_RECORD_TPCW
        pTxnRcrd); //support for TPC-W

        int WriteCtrlRecToLog(BYTE SubType,
        LPTSTR lpStr, DWORD dwLen);

        void CloseTransactionLogFile(void);

        PTXN_RECORD_HEADER
        GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
        PTXN_RECORD_HEADER
        GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
        bSkipCtrlRecs = FALSE);

        int Sort(void);
        PTXN_RECORD_HEADER
        GetSortedRecord();

        inline BOOL IsSorted(void) { return
        bLogSorted; };
        inline JULIAN_TIME BeginTS(void)
        { return BeginTxnTS; };
        inline JULIAN_TIME EndTS(void)

```

```

{ return EndTxnTS; };
        inline int RecordCount(void) { return
iRecCount; };
};

class CTXNLOG_ERR : public CBaseErr
{
public:
        enum CTXNLOG_ERRS
        {
                ERR_BAD_FILE_FORMAT,
                // "File format is invalid."

                ERR_UNKNOWN_LOG_VERSION, // "Log
file version is unknown."

                ERR_BROKEN_LOG_FILE,
                // "Log file is broken."

                ERR_LOG_NOT_SORTED,
                // "Log file is not sorted"

                ERR_INVALID_TIME_SEQ,
                // "Internal Error: Record Time Sequence
invalid."
        };

        CTXNLOG_ERR(int iErr) :
        CBaseErr(iErr) {};

        int ErrorType() {return
        ERR_TYPE_TXNLOG;};
        char *ErrorTypeStr() { return "TXN
        LOG"; }

        char *ErrorText()
        {
                static char *szMsgs[] = {
                        "File format is
        invalid.",
                        "Log file version is
        unknown.",
                        "Log file is broken.",
                        "Log file is not
        sorted",
                        "Internal Error:
        Record Time Sequence invalid.",
                        ""
                };

                for(int i = 0; szMsgs[i][0]; i++)
                {
                        if ( m_idMsg == i )
                                break;
                }

                return(szMsgs[i][0] ?

```

```
szMsgs[i] : ERR_UNKNOWN);
};
```

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);
```

spinlock.h

```
/* FILE: SPINLOCK.H
Copyright 1997 Microsoft Corp., All rights reserved.
Source code licensed to Tandem Computers for Internal
use only. Redistribution of source or object files or
any derivative works is prohibited. By agreement, this
notice may not be removed.
Authors: Mike Parkes, Charles Levine, Philip Durr
Microsoft Corp.
*/
#ifdef _INC_Spinlock
const LONG LockClosed = 1;
const LONG LockOpen = 0;
```

```
/******
Spinlock and Semaphore locking.
This class provides a very conservative locking
scheme.
The assumption behind the code is that locks will
be held for a very short time. When a lock is taken a
memory location is exchanged. All other threads that want
this lock wait by spinning and sometimes sleeping on
a semaphore until it becomes free again. The only other choice
is not to wait at all and move on to do something else.
This module should normally be used in conjunction
with cache aligned memory in minimize cache line misses.
*****/
```

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

#ifdef _DEBUG
// Counters for debugging
volatile LONG TotalLocks;
volatile LONG TotalSleeps;
volatile LONG TotalSpins;
volatile LONG TotalWaits;
#endif

public:
Spinlock( void );

inline BOOL
ClaimLock( BOOL Wait = TRUE );
```

```
inline void ReleaseLock( void );
~Spinlock( void );
// Disabled operations.
Spinlock( const Spinlock &
Copy );
void operator=( const Spinlock &
Copy );

private:
// Private functions.
inline BOOL
ClaimSpinlock( volatile LONG *sl );
void WaitForLock( void );
void WakeAllSleepers( void );
};

/******
A guaranteed atomic exchange.
An attempt is made to claim the Spinlock. This
action is guaranteed to be atomic.
*****/

inline BOOL Spinlock::ClaimSpinlock( volatile
LONG *Spinlock )
{
#ifdef _DEBUG
InterlockedIncrement( (LPLONG) & TotalLocks );
#endif
return ( (*Spinlock) == LockOpen) &&
(InterlockedExchange( (LPLONG)Spinlock, LockClosed) ==
LockOpen);
}

/******
Claim the Spinlock.
Claim the lock if available else wait or exit.
*****/

inline BOOL Spinlock::ClaimLock( BOOL Wait )
{
if ( ! ClaimSpinlock( (volatile LONG*) &
```

```

m_Spinlock )
{
    if ( Wait )
        WaitForLock();
    return Wait;
}
return TRUE;
}

/*****
*****
*
* Release the Spinlock.
*
* Release the lock and if needed wakeup any
sleepers.
*
*****
*****/

inline void Spinlock::ReleaseLock( void )
{
    m_Spinlock = LockOpen;
    if ( Waiting > 0 )
        WakeAllSleepers();
}

#define _INC_Spinlock

#endif

```

rtetime.h

```

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

//FILE: RTETIME.H

#define MAX_JULIAN_TIME
0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME __int64

```

```

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

```

Database Stored Procedures

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
    END
END

```

```

-- 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

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 (repeatableread)
WHERE  i_id     = @li_id
-----
-- update stock values
-----
UPDATE stock
SET  s_ytd      = s_ytd + @li_qty,
     @s_quantity = s_quantity - @li_qty +
     CASE WHEN (s_quantity - @li_qty < 10)
THEN 91 ELSE 0 END,
     s_order_cnt = s_order_cnt + 1,
     s_remote_cnt = s_remote_cnt +
     CASE WHEN (@li_s_w_id = @w_id)
THEN 0 ELSE 1 END,
     @s_data     = s_data,
     @s_dist     = CASE @d_id
    WHEN 1 THEN s_dist_01
    WHEN 2 THEN s_dist_02
    WHEN 3 THEN s_dist_03
    WHEN 4 THEN s_dist_04
    WHEN 5 THEN s_dist_05
    WHEN 6 THEN s_dist_06
    WHEN 7 THEN s_dist_07
    WHEN 8 THEN s_dist_08
    WHEN 9 THEN s_dist_09

```

```

    WHEN 10 THEN s_dist_10
END
WHERE s_i_id   = @li_id AND
     s_w_id    = @li_s_w_id
-----
-- if there actually is a stock (and item) with these ids, go to
work
-----
IF (@@rowcount > 0)
BEGIN
-----
-- insert order_line data (using data from item and stock)
-----
INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @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
SELECT @commit_flag = 0
END
-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,

```

```

@c_discount = c_discount,
@c_credit = c_credit,
@c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
c_w_id = @w_id AND
c_d_id = @d_id

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,
@d_id,
@w_id,
@c_id_local,
0,
@o_ol_cnt,
@o_all_local,
@o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
@d_id,
@w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM warehouse WITH (repeatableread)
WHERE w_id = @w_id

IF (@commit_flag = 1)

COMMIT TRANSACTION n
ELSE
-----
-- all that work for nuthn!!!
-----
ROLLBACK TRANSACTION n

-----
-- return order data to client
-----
SELECT @w_tax,
@d_tax,
@o_id,
@c_last,
@c_discount,
@c_credit,
@o_entry_d,
@commit_flag

```

```

END
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

null-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
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 = 'dzKoCOBqbc3yu',
        @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 - @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(ORDER GROUP)
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

tpcc_neworder_new.sql

-----
-- File: TPCC_NEWORDER_NEW.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.68 --
-- Copyright Microsoft, 2006 --

```

```

--
-- This acid stored procedure implements the neworder --
-- transaction. It outputs timestamps at the --
-- beginning of the transaction, before the commit --
-- delay, and after the commit.
--
-----
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_neworder_new' )
    DROP PROCEDURE tpcc_neworder_new
GO

-- neworder_new v2.5 6/23/05 PeterCa
-- 1q stock/order_line/client. upd district & ins neworder.
-- cust/warehouse select together, ins order separate
-- uses rownumber to distinct w any transform
-- uses in-memory sort for distinct on iid,wid
-- uses charindex
-- will rollback if (@i_idX,@s_w_idX pairs not unique) OR
(@i_idX not unique).

CREATE PROCEDURE tpcc_neworder_new
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1
smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2
smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @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 +
CASE WHEN
(info.w_id = @w_id) THEN 0 ELSE 1 END

```

```

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

```

```

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 uo1(iid,wid,lino,qty)
) AS o1(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

```

version.sql

```

-----
--
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Returns version level of TPC-C stored procs
--
-- Always update the return value of this proc for
-- any interface changes or 'must have' bug fixes.
--
-- The value returned by this SP defines the
-- 'interface level', which must match between the
-- stored procs and the client code. The
-- interface level may be down rev from the
-- current kit. This indicates that the interface
-- hasn't changed since that version.
--
-- Interface Level: 4.20.000
-----
USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name =
'tpcc_version' )

```

```

DROP PROCEDURE tpcc_version
GO

CREATE PROCEDURE tpcc_version
AS
DECLARE @version char(8)

BEGIN
SELECT @version = '4.20.000'

SELECT @version AS 'Version'
END
GO

```

Appendix B: Database Design Scripts

Database Build Source Code

backup.sql

```

-----
-- File: BACKUP.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
-----

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
        CONVERT(VARCHAR(30),@startdate,
21)

DUMP DATABASE tpcc TO tpcc1, tpcc2, tpcc3, tpcc4, tpcc5,
tpcc6, tpcc7, tpcc8, tpcc9, tpcc10, tpcc11, tpcc12, tpcc13, tpcc14,
tpcc15, tpcc16, tpcc17, tpcc18, tpcc19, tpcc20 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

```

createdb.sql

```

-----
-- File: CREATEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
-----

```

```

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 = 'F:\MSSQL_tpcc_root.mdf',
  SIZE = 8MB,
  FILEGROWTH = 0),
FILEGROUP MSSQL_misc_fg
( NAME = MSSQL_misc1,
  FILENAME = 'C:\mp\m1\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc2,
  FILENAME = 'C:\mp\m2\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc3,
  FILENAME = 'C:\mp\m3\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc4,
  FILENAME = 'C:\mp\m4\',

```

```

  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc5,
  FILENAME = 'C:\mp\m5\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc6,
  FILENAME = 'C:\mp\m6\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc7,
  FILENAME = 'C:\mp\m7\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc8,
  FILENAME = 'C:\mp\m8\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc9,
  FILENAME = 'C:\mp\m9\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc10,
  FILENAME = 'C:\mp\m10\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc11,
  FILENAME = 'C:\mp\m11\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc12,
  FILENAME = 'C:\mp\m12\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc13,
  FILENAME = 'C:\mp\m13\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_misc14,
  FILENAME = 'C:\mp\m14\',
  SIZE = 125000MB,
  FILEGROWTH = 0),
FILEGROUP MSSQL_cs_fg
( NAME = MSSQL_cs1,
  FILENAME = 'C:\mp\c1\',
  SIZE = 230000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_cs2,
  FILENAME = 'C:\mp\c2\',
  SIZE = 230000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_cs3,
  FILENAME = 'C:\mp\c3\',
  SIZE = 230000MB,

```

```

( FILEGROWTH = 0),
  NAME = MSSQL_cs4,
  FILENAME = 'C:\mp\c4\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs5,
  FILENAME = 'C:\mp\c5\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs6,
  FILENAME = 'C:\mp\c6\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs7,
  FILENAME = 'C:\mp\c7\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs8,
  FILENAME = 'C:\mp\c8\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs9,
  FILENAME = 'C:\mp\c9\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs10,
  FILENAME = 'C:\mp\c10\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs11,
  FILENAME = 'C:\mp\c11\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs12,
  FILENAME = 'C:\mp\c12\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs13,
  FILENAME = 'C:\mp\c13\',
  SIZE = 230000MB,
( FILEGROWTH = 0),
  NAME = MSSQL_cs14,
  FILENAME = 'C:\mp\c14\',
  SIZE = 230000MB,
  FILEGROWTH = 0)
LOG ON
( NAME = MSSQL_tpcc_log1,
  FILENAME = 'E:',
  SIZE = 2000000MB,
  FILEGROWTH = 0),
( NAME = MSSQL_tpcc_log2,
  FILENAME = 'G:',
  SIZE = 300000MB,
  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

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

```

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_cs_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

-----
-- 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

```

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_c1' )
    DROP INDEX order_line.order_line_c1

CREATE UNIQUE CLUSTERED INDEX order_line_c1 ON
order_line(ol_w_id, ol_d_id, ol_o_id, ol_number)
    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

```

idxordnc.sql

```

-----
-- File: IDXORDNC.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-- Creates non-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_nc1' )
    DROP INDEX orders.orders_nc1

CREATE INDEX orders_nc1 ON orders(o_w_id, o_d_id, o_c_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

```

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_c1' )
    DROP INDEX warehouse.warehouse_c1

CREATE UNIQUE CLUSTERED INDEX warehouse_c1 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

```

removedb.sql

```

-----
-- File: REMOVEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.68
-- Copyright Microsoft, 2006
--
-----
USE master
GO

-- remove any existing database and backup files
-----
EXEC sp_dbremove tpcc, dropdev
GO

EXEC sp_dropdevice 'tpcc1'
EXEC sp_dropdevice 'tpcc2'
EXEC sp_dropdevice 'tpcc3'
EXEC sp_dropdevice 'tpcc4'
EXEC sp_dropdevice 'tpcc5'
EXEC sp_dropdevice 'tpcc6'
EXEC sp_dropdevice 'tpcc7'
EXEC sp_dropdevice 'tpcc8'
EXEC sp_dropdevice 'tpcc9'
EXEC sp_dropdevice 'tpcc10'
EXEC sp_dropdevice 'tpcc11'
EXEC sp_dropdevice 'tpcc12'
EXEC sp_dropdevice 'tpcc13'
EXEC sp_dropdevice 'tpcc14'

```

```
EXEC sp_dropdevice 'tpcc15'
EXEC sp_dropdevice 'tpcc16'
EXEC sp_dropdevice 'tpcc17'
EXEC sp_dropdevice 'tpcc18'
EXEC sp_dropdevice 'tpcc19'
EXEC sp_dropdevice 'tpcc20'
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
```

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_cs_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_misc_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_cs_fg
go

```

VerifyTpccLoad_2.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 ''
GO

```

```

USE tpcc
GO

IF EXISTS (SELECT name
           FROM sysobjects
           WHERE name = 'TPCC_INFO' AND
                 type = 'U')
DROP TABLE TPCC_INFO
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

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_HISTORY = COUNT_BIG(*)
FROM history

SELECT @NUM_STOCK = COUNT_BIG(*)
FROM stock

SELECT @NUM_ORDER_LINE = COUNT_BIG(*)
FROM order_line

SELECT @NUM_NEW_ORDER = COUNT_BIG(*)
FROM new_order

--- 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

```

backupdev.sql

```

-----
-- File: BACKUPDEV.SQL --
-- Microsoft TPC-C Benchmark Kit Ver. 4.68 --
-- Copyright Microsoft, 2006 --
-----

USE master
GO

-----
-- create backup devices
-----
EXEC sp_addumpdevice 'disk','tpcc1','c:\mp\nb1\tpcc1.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc2','c:\mp\nb2\tpcc2.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc3','c:\mp\nb3\tpcc3.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc4','c:\mp\nb4\tpcc4.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc5','c:\mp\nb5\tpcc5.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc6','c:\mp\nb6\tpcc6.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc7','c:\mp\nb7\tpcc7.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc8','c:\mp\nb8\tpcc8.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc9','c:\mp\nb9\tpcc9.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc10','c:\mp\nb10\tpcc10.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc11','c:\mp\nb11\tpcc11.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc12','c:\mp\nb12\tpcc12.dmp'
GO

```

```
EXEC sp_addumpdevice 'disk','tpcc13','c:\mp\nb13\tpcc13.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc14','c:\mp\nb14\tpcc14.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc15','c:\mp\nb15\tpcc15.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc16','c:\mp\nb16\tpcc16.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc17','c:\mp\nb17\tpcc17.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc18','c:\mp\nb18\tpcc18.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc19','c:\mp\nb19\tpcc19.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcc20','c:\mp\nb20\tpcc20.dmp'
GO
```

delwh.sql

```
use tpcc
go
delete from warehouse where w_id > 54349
go
select count(*) from warehouse
go
select * from warehouse where w_id > 54349
go
print ''
select convert(char(30), getdate(),9)
print ''
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_cs_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_c1' )
DROP INDEX history.history_c1

CREATE UNIQUE CLUSTERED INDEX history_c1 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_c1' )
DROP INDEX new_order.new_order_c1

CREATE UNIQUE CLUSTERED INDEX new_order_c1 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
```

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_c1' )
  DROP INDEX orders.orders_c1

CREATE UNIQUE CLUSTERED INDEX orders_c1 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_c1' )
  DROP INDEX stock.stock_c1

```

```

CREATE UNIQUE CLUSTERED INDEX stock_c1 ON
stock(s_i_id, s_w_id)
  ON MSSQL_cs_fg

```

```

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

newDB.sql

```

use master
go
alter database tpcc set page_verify none
go
use tpcc
go
drop index orders.orders_nc1
go
sp_dboption tpcc,'torn page detection', false
go

```

restore.sql

```

-----
--      File:  RESTORE.SQL                          --
--      Microsoft TPC-C Benchmark Kit Ver. 4.68      --
--      Copyright Microsoft, 2006                    --
--                                                    --
-----
DECLARE @startdate      DATETIME,
        @enddate        DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,
21)

RESTORE DATABASE tpcc FROM tpcc1, tpcc2, tpcc3, tpcc4,
tpcc5, tpcc6, tpcc7, tpcc8, tpcc9, tpcc10, tpcc11, tpcc12, tpcc13,
tpcc14, tpcc15, tpcc16, tpcc17, tpcc18, tpcc19, tpcc20 WITH
stats = 1

SELECT @enddate = GETDATE()
SELECT 'End date:',

```

```

CONVERT(VARCHAR(30),@enddate, 21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
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

```

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

```

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

```

Database Loader Source Code

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, TPCCLDR_ARGS
 *pargs)
{
  int i;
  char *ptr;

#ifdef DEBUG

```

```

  printf("[%ld]DBG: Entering GetArgsLoader()\n", (int)
  GetCurrentThreadId());
#endif

  /* init args struct with some useful values */
  pargs->server = SERVER;
  pargs->user = USER;
  pargs->password = PASSWORD;
  pargs->database = DATABASE;
  pargs->batch = BATCH;
  pargs->num_warehouses = UNDEF;
  pargs->tables_all = TRUE;
  pargs->table_item =
FALSE;
  pargs->table_warehouse =
FALSE;
  pargs->table_customer =
FALSE;
  pargs->table_orders =
FALSE;
  pargs->loader_res_file =
LOADER_RES_FILE;
  pargs->log_path =
  LOADER_LOG_PATH;
  pargs->pack_size =
DEFLDPACKSIZE;
  pargs->starting_warehouse =
DEF_STARTING_WAREHOUSE;
  pargs->build_index =
BUILD_INDEX;
  pargs->index_order =
INDEX_ORDER;
  pargs->index_script_path =
INDEX_SCRIPT_PATH;
  pargs->scale_down =
SCALE_DOWN;

  /* 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])
    {

```

<pre> case '?': /* Fall through */ GetArgsLoaderUsage(); break; case 'D': ptr+2; pargs->database = break; case 'P': ptr+2; pargs->password = break; case 'S': ptr+2; pargs->server = break; case 'U': pargs->user = ptr+2; break; case 'b': pargs->batch = atol(ptr+2); break; case 'W': >num_warehouses = atol(ptr+2); break; case 's': >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) </pre>	<pre> 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(); </pre>	<pre> exit(-1); break; } } /* check for required args */ if (pargs->num_warehouses == UNDEF) { printf("Number of Warehouses is required\n"); exit(-2); } return; } } //===== // // Function name: GetArgsLoaderUsage // //===== void GetArgsLoaderUsage() { #ifdef DEBUG printf("[%ld]DBG: Entering GetArgsLoaderUsage()\n", (int) GetCurrentThreadId()); #endif printf("TPCCLDR:\n\n"); printf("Parameter Default\n"); printf("-----\n"); printf("-W Number of Warehouses to Load Required \n"); printf("-S Server %s\n", SERVER); printf("-U Username %s\n", USER); printf("-P Password %s\n", PASSWORD); printf("-D Database %s\n", DATABASE); printf("-b Batch Size %ld\n", (long) BATCH); printf("-p TDS packet size %ld\n", (long) DEFLDPACKSIZE); printf("-L Loader BCP Log Path %s\n", LOADER_LOG_PATH); printf("-f Loader Results Output Filename </pre>
--	--	---

```

%s\n", LOADER_RES_FILE);
    printf("-s Starting Warehouse
%d\n", (long) DEF_STARTING_WAREHOUSE);
    printf("-i Build Option (data = 0, data and index = 1)
%d\n", (long) BUILD_INDEX);
    printf("-o Cluster Index Build Order (before = 1, after
= 0) %d\n", (long) INDEX_ORDER);
    printf("-c Build Scaled Database (normal = 0, tiny = 1)
%d\n", (long) SCALE_DOWN);
    printf("-d Index Script Path
%s\n", INDEX_SCRIPT_PATH);
    printf("-t Table to Load          all
tables \n");
    printf(" [item|warehouse|customer|orders]\n");
    printf(" Notes: \n");
    printf(" - the '-t' parameter may be included multiple times
to \n");
    printf(" specify multiple tables to be loaded \n");
    printf(" - 'item' loads ITEM table \n");
    printf(" - 'warehouse' loads WAREHOUSE, 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);
}

```

restore.vbs

```

'-----
'--- FILE: RESTORE.VBS
'--- Microsoft TPC-C Kit Ver. 4.68
'--- Copyright Microsoft, 2001, 2002, 2006
'--- All Rights Reserved
'---
'--- PURPOSE: This module executes a database restore
'---
'-----
'-----
'--- set the kit version variable for later display
'-----
Kit_Version = " 4.68"
'-----
'--- open an windows scripting object
'-----

```

```

set WshShell = CreateObject("WScript.Shell")
'-----
'--- set up windows scripting argument collection
'-----
Set objArgs = wScript.Arguments
'-----
'--- display a banner message
'-----
wScript.Echo
"*****"
wScript.Echo "*"
wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & "*"
wScript.Echo "*"
wScript.Echo "*" Restore tpcc database
*"
wScript.Echo "*"
wScript.Echo
"*****"
'-----
'--- define function to check for any error messages
'-----
Function CheckSQLOutput(SQL_Out)
    ErrorFlag = 0
    Set SQL_fso =
CreateObject("Scripting.FileSystemObject")
    Set SQL_Out_File =
SQL_fso.OpenTextFile(SQL_Out,1)
    Do While SQL_Out_File.AtEndOfStream <> True
        SQL_Line = SQL_Out_File.ReadLine
        'first check to see if the output contains a
message about the login password
        If InStr(SQL_Line, "Login failed") Then
            'display the messages and get
            ErrorFlag = 1
            wScript.Echo "The login for
userid 'sa' failed."
            wScript.Echo "Please restart
SETUP with the correct password."
            Else
                If InStr(SQL_Line, "Msg")
Then
                    'find out where the "Msg" indicator is in the
line
                    LocMsg = InStr(SQL_Line, "Msg")
                    'find out where the comma is after the error
code
                    LocComma = InStr(SQL_Line, ",")
                    'now isolate the error code
                    ErrorCode = Mid(SQL_Line, (LocMsg + 4),
(LocComma - (LocMsg + 4)))

```

```

Select Case ErrorCode
Case "15069"
    ErrorFlag = 1
    wScript.Echo "One or more users are
using the database."
    wScript.Echo "The requested operation
cannot be completed."
Case "3201"
    ErrorFlag = 1
    wScript.Echo "Cannot open backup
device."
    wScript.Echo "Device error or device
off-line."
    wScript.Echo "SQL Server Error 3201."
    wScript.Echo "See the SQL Server error
log for more details."
End Select
End If
End If
Loop
SQL_Out_File.Close
CheckSQLOutput = ErrorFlag
End Function
'-----
'--- end function
'-----
'-----
'--- open a file system object
'-----
Set fs = CreateObject("Scripting.FileSystemObject")
'-----
'--- grab the current directory value
'-----
SetupDirectory = WshShell.CurrentDirectory & "\
'-----
'--- now calculate the other directories
'-----
ScriptDirectory = SetupDirectory & "SCRIPTS\
LogDirectory = SetupDirectory & "LOGS\
'-----
'--- check to see if the user passed in the server name and sa
password
'-----
Select Case objArgs.Length
Case 0
'-----
'--- the user did not pass us anything
'--- grab the Computer Name from
Windows
'-----
ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%
")
'-----

```

```

name          '--- prompt the user to confirm the server
              '-----
              ServerName = InputBox("Enter your
server name",Test_Name,ServerName)
              Do While ServerName = ""
                  rc = MsgBox ("You must enter
a valid server name.",21)
              If rc = 2 Then
                  wScript.Echo ""
                  wScript.Echo "TPC-
C Setup cancelled by user."
                  wScript.Quit
              End If
              ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%
")
              ServerName = InputBox("Enter
your server name","Database Server Name",ServerName)
              Loop
              '-----
              '--- prompt the user for the sa password
              '-----
              saPassword = InputBox("Enter the 'sa'
password",Test_Name)
              Case 1
              '-----
              '--- the user passed 1 argument, so assume
it is the server name
              '-----
              ServerName = objArgs(0)
              '-----
              '--- prompt the user for the sa password
              '-----
              saPassword = InputBox("Enter the 'sa'
password",Test_Name)
              Case 2
              '-----
              '--- the user passed 2 arguments, so try to
use them
              '-----
              ServerName = objArgs(0)
              saPassword = objArgs(1)
          End Select
          '-----
          '--- now that we have all the variables filled in, let's get to work
          '-----
          If fs.FileExists(LogDirectory & "restore.log") Then
              fs.DeleteFile LogDirectory & "restore.log"
          End If
          Wscript.Echo "Restoring database from backup..."
          Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -

```

```

S" & ServerName & " -e -i" & ScriptDirectory &
NumberWarehouses & ".war\database\restore.sql -o" &
LogDirectory & "restore.log")
Do While oExec.Status = 0
    wScript.Sleep 100
Loop
rc = CheckSQLOutput(LogDirectory & "restore.log")
If rc <> 0 Then
    wScript.Quit
End If
wScript.Echo ""
wScript.Echo
*****
wScript.Echo "*"
wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & "        *"
wScript.Echo "*"
wScript.Echo "*" Database restore complete.
*"
wScript.Echo "*"
wScript.Echo
*****

```

setup.vbs

```

'-----
'--- FILE:  SETUP.VBS
'---      Microsoft TPC-C Kit Ver. 4.68
'---      Copyright Microsoft, 2001, 2002, 2006, 2004, 2006
'---      All Rights Reserved
'---
'--- PURPOSE:  This module performs the tasks to create and
populate a TPC-C database
'---
'-----
'--- set the kit version variable for later display
'-----
Kit_Version = " 4.68"
SQL2K_Kit_Version = " 4.55"
'-----
'--- open an windows scripting object
'-----
Set WshShell = CreateObject("WScript.Shell")
'-----
'--- set up windows scripting argument collection
'-----

```

```

Set ObjArgs = wScript.ARGUMENTS
'-----
'--- grab the platform, ia64, x86, from the environment variables
'-----
Platform =
LCase(Left(WshShell.ExpandEnvironmentStrings("%PROCESS
OR_IDENTIFIER%"), 4))
Select Case Platform
    Case "ia64"
        Platform = "IA64"
    Case Else
        Platform = "X86"
End Select
'-----
'--- grab the processor architecture. This is to determine if the
'--- user is trying to run in 32-bit emulation on a 64-bit machine.
'--- if that is the case, then display a message and exit.
'-----
Proc_Architecture =
WshShell.ExpandEnvironmentStrings("%PROCESSOR_ARCHI
TECTURE%")
If Platform = "IA64" And Proc_Architecture = "x86" Then
    wScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    wScript.Echo
    "!!                                 !!"
    wScript.Echo "!! You are attempting to run this SETUP in
the 32-bit (WOW) emulation !!"
    wScript.Echo "!! mode on an ia64 system. Please restart the
SETUP in a native    !!"
    wScript.Echo "!! 64-bit
environment.                                !!"
    wScript.Echo
    "!!                                 !!"
    wScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    wScript.Quit
End If
'-----
'--- before we go any further, make sure that
'--- we are running Windows Scripting Host 5.6
'--- or higher
'-----
If wScript.Version < 5.6 Then
    wScript.Echo
    "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    wScript.Echo
    "!!                                 !!"
    wScript.Echo "!! You do not have the proper version of the
Windows Scripting Host    !!"
    wScript.Echo "!! installed. Please install the latest
Windows Scripting Host from    !!"

```

```

wScript.Echo "!! ..tools\wsh\scripen.exe and restart
setup.      !!"
wScript.Echo
"!!                !!"
wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
wScript.Quit
End If
'-----
'--- display banner message
'-----
wScript.Echo
"*****"
wScript.Echo "*"                "*"
wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & "                *"
wScript.Echo "*"                "*"
wScript.Echo "*" Database Setup
*"
wScript.Echo "*"                "*"
wScript.Echo
"*****"
"*****"
'-----
'--- Initialize an array of the TPC-C table names
'-----
Dim TableArray(9)
TableArray(0) = "warehouse"
TableArray(1) = "district"
TableArray(2) = "customer"
TableArray(3) = "history"
TableArray(4) = "new_order"
TableArray(5) = "orders"
TableArray(6) = "order_line"
TableArray(7) = "item"
TableArray(8) = "stock"
TableArray(9) = "tpcclldr"
'-----
'--- Initialize an array of the TPC-C build log file names
'-----
Dim LogFileArray(21)
LogFileArray(0) = "version.log"
LogFileArray(1) = "removedb.log"
LogFileArray(2) = "createdb.log"
LogFileArray(3) = "tables.log"
LogFileArray(4) = "dbopt1.log"
LogFileArray(5) = "idxordcl.log"
LogFileArray(6) = "idxitmcl.log"
LogFileArray(7) = "idxwarcl.log"
LogFileArray(8) = "idxcuscl.log"
LogFileArray(9) = "idxnodcl.log"
LogFileArray(10) = "idxdiscl.log"
LogFileArray(11) = "idxstkcl.log"

```

```

LogFileArray(12) = "idxodcl.log"
LogFileArray(13) = "idxcusnc.log"
LogFileArray(14) = "idxhiscl.log"
LogFileArray(15) = "idxordnc.log"
LogFileArray(16) = "bulkload.log"
LogFileArray(17) = "dbopt2.log"
LogFileArray(18) = "nurand_load.log"
LogFileArray(19) = "backupdev.log"
LogFileArray(20) = "backupdev.log"
LogFileArray(21) = "verifyload.log"
'-----
'--- open a file system object
'-----
Set fs = CreateObject("Scripting.FileSystemObject")
'-----
'--- first see if the user passed a "?" as the first parameter
'--- if they did, then show the usage data
'-----
If ObjArgs.Length > 0 Then
    If ObjArgs(0) = "?" or ObjArgs(0) = "/" Then
        Call ShowUsage
    End If
End If
'--- get the user passed in parameters
If WScript.Arguments.Named.Exists("S") Then
    ServerName = WScript.Arguments.Named.Item("S")
    If ServerName = "" Then
        ServerName =
        WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    End If
    flgServerName = 1
Else
    ServerName =
    WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%")
    flgServerName = 0
End If
SQLUserID = WScript.Arguments.Named.Item("U")
If SQLUserID <> "" Then
    flgSQLUserID = 1
Else
    flgSQLUserID = 0
End If
SQLPassword = WScript.Arguments.Named.Item("P")
If SQLPassword <> "" Then
    flgSQLPassword = 1
Else
    flgSQLPassword = 0
End If
If SQLPassword = "BLANK" Then
    flgSQLPassword = 1
SQLPassword = ""
End If

```

```

NumberWarehouses = WScript.Arguments.Named.Item("W")
If NumberWarehouses <> "" Then
    flgNumberWarehouses = 1
Else
    flgNumberWarehouses = 0
End If
BuildOption = WScript.Arguments.Named.Item("B")
If BuildOption <> "" Then
    'validate the build option the user passed in
    BuildOption = LCase(BuildOption)
    Select Case BuildOption
        Case
            "full","builddb","objects","objectsfull","bulkload","bulkloadfull",
            "backup"
                flgBuildOption = 1
        Case Else
                flgBuildOption = 0
    End Select
Else
    flgBuildOption = 0
End If
DatabaseType = WScript.Arguments.Named.Item("D")
If DatabaseType <> "" Then
    DatabaseType = LCase(DatabaseType)
    Select Case DatabaseType
        Case "normal","scale_down"
            If DatabaseType = "normal" Then
                DatabaseType = 0
            Else
                DatabaseType = 1
            End If
            flgDatabaseType = 1
        Case Else
            flgDatabaseType = 0
    End Select
Else
    flgDatabaseType = 0
End If
UnattendedBuild = WScript.Arguments.Named.Item("V")
If UnattendedBuild <> "" Then
    UnattendedBuild = LCase(UnattendedBuild)
    Select Case UnattendedBuild
        Case "true","false"
            flgUnattendedBuild = 1
        Case Else
            flgUnattendedBuild = 0
    End Select
Else
    flgUnattendedBuild = 0
End If
'--- if something is missing, prompt the user for it
If ServerName = 0 Then
    ServerName = Get userInput("ServerName")
    If ServerName = "" Then

```

```

ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%
")
End If
End If
If flgSQLUserID = 0 Then
    SQLUserID = Get userInput("SQLUserID")
End If
If flgSQLPassword = 0 Then
    SQLPassword = Get userInput("SQLPassword")
End If
If flgNumberWarehouses = 0 Then
    NumberWarehouses = Get userInput("NumberWarehouses")
End If
If flgBuildOption = 0 Then
    BuildOption = Get userInput("BuildOption")
End If
If flgDatabaseType = 0 Then
    DatabaseType = (Get userInput("DatabaseType"))
End If
If flgUnattendedBuild = 0 Then
    UnattendedBuild = Get userInput("UnattendedBuild")
End If
If SQLPassword = "BLANK" Then
    SQLPassword = ""
End If
'-----
'--- if the user specified a scale down database, then show
'--- them the warning message
'-----
If DatabaseType = 1 Then
    MsgBox "WARNING!" & Chr(13) & "The Scale_Down
option is to be used for functional testing only." _
    & Chr(13) & "The use of this option will not
produce a valid TPC-C result." _
    vbExclamation, "Scale-Down Warning"
End If
'-----
'--- before we start to do anything, verify the input
'-----
Select Case BuildOption
    Case "full"
        strBuildOpt = "Full build"
    Case "builddb"
        strBuildOpt = "Build database only"
    Case "objects"
        strBuildOpt = "Install stored procedures only"
    Case "objectsfull"
        strBuildOpt = "Install stored procedures and complete
build process"
    Case "bulkload"
        strBuildOpt = "Load data only"
    Case "bulkloadfull"
        strBuildOpt = "Load data and complete build process"

```

```

Case "backup"
    strBuildOpt = "Backup database"
End Select
If DatabaseType = 1 Then
    strDBType = "Scale Down"
Else
    strDBType = "Normal"
End If
If UnattendedBuild = "false" Then
    rc = MsgBox("The following options will be used." &
Chr(13) & Chr(10) & Chr(13) & Chr(10) _
    & "Database Server Name:
" & ServerName & Chr(13) & Chr(10) _
    & Chr(13) & Chr(10) _
    & "SQL Server User ID:
" & SQLUserID & Chr(13) & Chr(10) _
    & Chr(13) & Chr(10) _
    & "SQL Server Password:
" & SQLPassword & Chr(13) & Chr(10) _
    & Chr(13) & Chr(10) _
    & "Number of Warehouses:
" & NumberWarehouses & Chr(13) & Chr(10) _
    & Chr(13) & Chr(10) _
    & "Build Option:
" & strBuildOpt & Chr(13) & Chr(10) _
    & Chr(13) & Chr(10) _
    & "Build Type:
" & strDBType & Chr(13) & Chr(10) _
    & "
", 65, "TPC-C Setup Option Verification")
    If rc = 2 Then
        wScript.Echo ""
        wScript.Echo "TPC-C Setup cancelled by user."
        wScript.Quit
    End If
End If
'-----
'--- parse the ServerName to determine if this is a named instance
'-----
Slash_Loc = InStr(1,ServerName,"")
If Slash_Loc <> 0 Then
    SQLInstanceName = Right(ServerName,(LEN(ServerName) -
Slash_Loc))
Else
    SQLInstanceName = ""
End If
'-----
'--- now we need to figure out if this is SQL Server 2000 or SQL
Server 2005
'--- if this is being installed on SQL Server 2000, then abort the
load and
'--- direct the user to use the 4.55 kit which is SQL Server 2000
compliant
'-----

```

```

If SQLInstanceName = "" Then
    '--- check for default installations
    '--- SQL Server 2000 Default Instance
    strRegKey =
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLS
erver\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    '     SQLServerVersionRegKey =
WshShell.RegRead(strRegKey)
    ' End If
    '--- SQL Server 2005 Default Instance
    strRegKey =
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLS
erver\MSSQLServer\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    '     SQLServerVersionRegKey =
WshShell.RegRead(strRegKey)
    ' End If
'Else
    '--- SQL Server 2000 Named Instance
    'If
    CheckRegKey("HKEY_LOCAL_MACHINE\SOFTWARE\Micr
osoft\MSSQLServer\CurrentVersion\CurrentVersion") = True
    Then
        ' SQLServerVersionRegKey =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\MSSQLServer\CurrentVersion\CurrentVersion")
    'End If
    '--- SQL Server 2005 Named Instance
    strRegKey =
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsof
t SQL Server" & SQLInstanceName &
"\MSSQLServer\CurrentVersion\CurrentVersion"
    ' If CheckRegKey(strRegKey) = True Then
    '     SQLServerVersionRegKey =
WshShell.RegRead(strRegKey)
    ' End If
'End If
'If Left(SQLServerVersionRegKey,1) = "8" Then
    ' SQLServerVersion = "2000"
    ' wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    ' wScript.Echo "!!"
    ' wScript.Echo "!! The Microsoft TPC-C Benchmark Kit
Version" & Kit_Version & " is for use with !!"
    ' wScript.Echo "!! SQL Server 2005 only. If you require SQL
Server 2000 then you !!"
    ' wScript.Echo "!! must use the Microsoft TPC-C Benchmark
Kit Version" & SQL2K_Kit_Version & ". !!"
    ' wScript.Echo "!!"
    ' wScript.Echo
"!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
    ' wScript.Quit
'Else

```

```

    SQLServerVersion = "2005"
'End If
'-----
'--- set the directory path names
'-----
SetupDirectory = WshShell.CurrentDirectory & "\"
ACIDDirectory = Left(SetupDirectory, (Len(SetupDirectory) -
6))
ScriptDirectory = SetupDirectory & "Scripts\"
UtilityDirectory = ScriptDirectory & "Utility\"
DDLDirectory = ScriptDirectory & NumberWarehouses &
".war\DDL\"
DMLDirectory = ScriptDirectory & "DML\"
DBDirectory = ScriptDirectory & NumberWarehouses &
".war\Database\"
'-----
'--- the log directory will be a concatenation
'--- of the number of customers and the date
'-----
LogDirectory = SetupDirectory & "Logs\" &
NumberWarehouses & "_Warehouses_" & Month(Now) & "-" &
Day(Now) & "-" & Year(Now) & "-"
'-----
'--- now figure out if a previous directory exists with this
'--- customer and data combination. If not, the append a
'--- 1 to the directory name, else append with the next
'--- number in sequence.
'-----
Dir_Flag = 0
Directory_Index = 1
Do While Dir_Flag = 0
    if (fs.FolderExists(LogDirectory & Directory_Index))
Then
        Directory_Index = Directory_Index + 1
    Else
        LogDirectory = LogDirectory &
Directory_Index
        Dir_Flag = 1
    End If
Loop
fs.CreateFolder(LogDirectory)
LogDirectory = LogDirectory & "\"
'-----
'--- initialize the BuildSteps.log file and output all of the
'--- particulars
'-----
BuildStepLogFile = LogDirectory & "BuildSteps.log"
Call WriteBuildLog("Begin TPC-C Setup Process", "")
Call
WriteBuildLog("====="
=====
=====, "")
Call WriteBuildLog("SQL Server Name = " & ServerName, "")
Call WriteBuildLog("System Platform = " & Platform, "")

```

```

Call WriteBuildLog("System Architecture = " &
Proc_Architecture, "")
Call WriteBuildLog("Number of Processors = " &
NumberofProcessors, "")
Call WriteBuildLog("SQL Server Version = " &
SQLServerVersionRegKey, "")
Call WriteBuildLog("SQL Server User ID = " & SQLUserID, "")
If SQLPassword = "" Then
    Call WriteBuildLog("SQL Server Password =
<Blank>", "")
Else
    Call WriteBuildLog("SQL Server Password =
*****", "")
End If
Call WriteBuildLog("Number of Warehouses = " &
NumberWarehouses, "")
Call WriteBuildLog("Build Option = " & BuildOption, "")
Call WriteBuildLog("Database Type = " & DatabaseType, "")
Call WriteBuildLog("Unattended Build = " & UnattendBuild, "")
Call WriteBuildLog("Setup Directory = " & SetupDirectory, "")
Call WriteBuildLog("ACID Directory = " & ACIDDirectory, "")
Call WriteBuildLog("Script Directory = " & ScriptDirectory, "")
Call WriteBuildLog("Utility Directory = " & UtilityDirectory, "")
Call WriteBuildLog("DDL Directory = " & DDLDirectory, "")
Call WriteBuildLog("DML Directory = " & DMLDirectory, "")
Call WriteBuildLog("Database Directory = " & DBDirectory, "")
Call WriteBuildLog("Log Directory = " & LogDirectory, "")
Call
WriteBuildLog("====="
=====
=====, "")
'-----
'--- now that we have all the variables filled in, let's get to work
'--- cleanup any old .err files
'-----
For i = 0 To 9
    If fs.FileExists(LogPath & TableArray(i) & ".err") Then
        fs.DeleteFile LogPath & TableArray(i) & ".err"
    End If
Next
For i = 0 To 21
    If fs.FileExists(LogPath & LogFileArray(i)) Then
        fs.DeleteFile LogPath & LogFileArray(i)
    End If
Next
wScript.Echo FormatDateTime(Now, 0) & " ==> Checking
connectivity to SQL Server..."
Call WriteBuildLog("Verifying SQL Server connectivity", "")
Set oExec = WshShell.Exec("osql -U" & SQLUserID & "-P" &
SQLPassword & "-S" & ServerName & "-e -i" &
UtilityDirectory & "version.sql -o" & LogDirectory &
"version.log")
Do While oExec.Status = 0
    wScript.Sleep 100

```

```

Loop
rc = CheckSQLOutput(LogDirectory & "version.log")
If rc <> 0 Then
    Call WriteBuildLog("Verifying SQL Server
connectivity", "Step failed! - Check version.log")
    wScript.Echo FormatDateTime(Now, 0) & " ==> Could not
connect to SQL Server! Check version.log."
    wScript.Quit
End If
Call WriteBuildLog("SQL Server connectivity verified", "")
wScript.Echo ""
'-----
'--- okay, let's do it
'-----
If (BuildOption = "full" Or BuildOption = "builddb") Then
    wScript.Echo FormatDateTime(Now, 0) & " ==> Removing
any existing TPC-C database and backup devices..."
    Call WriteBuildLog("Removing any existing TPC-C
database and backup devices", "")
    wScript.Echo ""
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& "-P" & SQLPassword & "-S" & ServerName & "-e -i" &
DBDirectory & "RemoveDB.sql -o" & LogDirectory &
"RemoveDB.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"RemoveDB.log")
    If rc <> 0 Then
        Call WriteBuildLog("Removing any
existing TPC-C database and backup devices", "Step failed! -
Check RemoveDB.log")
        wScript.Echo FormatDateTime(Now, 0) &
" ==> Removing existing TPC-C database failed! Check
RemoveDB.log."
        wScript.Quit
    End If
    Call WriteBuildLog("Any existing TPC-C database
and backup devices removed", "")
    wScript.Echo FormatDateTime(Now, 0) & " ==>
Building database files and database..."
    Call WriteBuildLog("Create TPC-C database", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& "-P" & SQLPassword & "-S" & ServerName & "-e -i" &
DBDirectory & "CreateDB.sql -o" & LogDirectory &
"CreateDB.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"CreateDB.log")
    If rc <> 0 Then
        Call WriteBuildLog("Create TPC-C

```



```

database","Step failed! - Check CreateDB.log")
    wScript.Echo FormatDateTime(Now,0) &
" ==> Creation of TPC-C database failed!! Check
CreateDB.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==>
TPC-C Database creation complete."
    Call WriteBuildLog("TPC-C database created", "")
End If
'-----
'--- build tables and stored procedures
'-----
If (BuildOption = "full" Or BuildOption = "objects" Or
BuildOption = "objectsfull") Then
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now,0) & " ==> Creating
TPC-C database tables..."
    Call WriteBuildLog("Create dynamic TPC-C
database tables", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DDLDirectory & "Tables.sql -o" & LogDirectory & "Tables.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "Tables.log")
    If rc <> 0 Then
        Call WriteBuildLog("Create TPC-C
database tables","Step failed! - Check Tables.log")
        wScript.Echo FormatDateTime(Now,0) &
" ==> Creation of TPC-C database tables failed!! Check
Tables.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==>
TPC-C database tables created."
    Call WriteBuildLog("TPC-C database tables
created", "")
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now,0) & " ==> Installing
TPC-C stored procedures..."
    Call WriteBuildLog("Install TPC-C stored
procedures", "")
    wScript.Echo "          New Order..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (New Order)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "neword.sql -o" & LogDirectory &
"SP_NewOrd.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop

```

```

    rc = CheckSQLOutput(LogDirectory &
"SP_NewOrd.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (New Order)","Step failed! - Check SP_NewOrd.log")
        wScript.Quit
    End If
    wScript.Echo "          New Order (New)..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (New Order (New))", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "TPCC_NEWORDER_NEW.SQL -o" &
LogDirectory & "SP_NewOrd_New.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_NewOrd_New.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (New Order (New))","Step failed! - Check
SP_NewOrd_New.log")
        wScript.Quit
    End If
    wScript.Echo "          Payment..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (Payment)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "payment.sql -o" & LogDirectory &
"SP_Payment.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Payment.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (Payment)","Step failed! - Check SP_Payment.log")
        wScript.Quit
    End If
    wScript.Echo "          Order Status..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (Order Status)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "ordstat.sql -o" & LogDirectory &
"SP_OrdStat.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_OrdStat.log")

```

```

    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (Order Status)","Step failed! - Check
SP_OrdStat.log")
        wScript.Quit
    End If
    wScript.Echo "          Delivery..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (Delivery)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "delivery.sql -o" & LogDirectory &
"SP_Delivery.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Delivery.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (Delivery)","Step failed! - Check SP_Delivery.log")
        wScript.Quit
    End If
    wScript.Echo "          Stock Level..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (Stock Level)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "stocklev.sql -o" & LogDirectory &
"SP_StockLev.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_StockLev.log")
    If rc <> 0 Then
        Call WriteBuildLog("Install TPC-C stored
procedures (Stock Level)","Step failed! - Check
SP_StockLev.log")
        wScript.Quit
    End If
    wScript.Echo "          Version (Internal)..."
    Call WriteBuildLog("    Install TPC-C stored
procedures (Version)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
DMLDirectory & "version.sql -o" & LogDirectory &
"SP_Version.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"SP_Version.log")
    If rc <> 0 Then

```

```

        Call WriteBuildLog("Install TPC-C stored
procedures (Version)","Step failed! - Check SP_Version.log")
    wScript.Quit
End If
wScript.Echo FormatDateTime(Now,0) & " ==>
TPC-C stored procedures installed."
    Call WriteBuildLog("TPC-C stored procedures
installed", "")
    wScript.Echo ""
End If
If (BuildOption = "full" Or BuildOption = "objectsfull" Or
BuildOption = "bulkload" Or BuildOption = "bulkloadfull")
Then
    wScript.Echo FormatDateTime(Now,0) & " ==> Setting
database options before load..."
    Call WriteBuildLog("Set pre-load database options
(DBOPT1)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
UtilityDirectory & "dbopt1.sql -o" & LogDirectory &
"Database_Options_1.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"Database_Options_1.log")
    If rc <> 0 Then
        Call WriteBuildLog("Set pre-load
database options (DBOPT1)","Step failed! - Check
Database_Options_1.log")
    End If
    wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==> Pre-
load database options set."
    Call WriteBuildLog("Pre-load database options
set", "")
    wScript.Echo ""
    '-----
    '--- before we start tpccldr.exe, check the registry
    '--- to ensure that the Shared Memory Protocol is off.
    '--- if it is on, store the setting so we can return
    '--- the system to the pre-tpccldr state.
    '-----
    If
CheckRegKey("HKEY_LOCAL_MACHINE\SOFTWARE\Micr
osoft\MSSQLServer\Client\SharedMemoryOn") = True Then
        SharedMemoryRegKey =
WshShell.RegRead("HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\MSSQLServer\Client\SharedMemoryOn")
        If SharedMemoryRegKey = 1 Then
            WshShell.RegWrite
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLS
erver\Client\SharedMemoryOn", 0, "REG_DWORD"
        End If

```

```

Else
    SharedMemoryRegKey = 0
End If
Call WriteBuildLog("Loading database and creating
indexes", "")
wScript.Echo FormatDateTime(Now,0) & " ==>
Loading database and creating indexes..."
wScript.Echo FormatDateTime(Now,0) & " ==>
(This runs in a separate, minimized window.)"
wScript.Echo ""
CMD_String = ""
wScript.Echo Platform
Select Case Platform
    Case "IA64"
        CMD_String = SetupDirectory &
"loader\bin\ia64\tpccldr.exe"
    Case Else
        CMD_String = SetupDirectory &
"loader\bin\x86\tpccldr.exe"
    End Select
CMD_String = CMD_String & " -S" & ServerName
CMD_String = CMD_String & " -U" & SQLUserID
CMD_String = CMD_String & " -P" & SQLPassword
CMD_String = CMD_String & " -W" &
NumberWarehouses
CMD_String = CMD_String & " -f" & LogDirectory
& "bulkload.log"
CMD_String = CMD_String & " -L" & LogDirectory
CMD_String = CMD_String & " -d" &
DDLDirectory
CMD_String = CMD_String & " -c" & DatabaseType
wScript.Echo CMD_String
oExec = WshShell.Run(CMD_String, 2, True)
If oExec <> 0 Then
    wScript.Echo FormatDateTime(Now,0) & " ==> The
TPCCCLR.EXE encountered an error."
    wScript.Echo FormatDateTime(Now,0) & " ==> Check the
TPCCCLR.ERR log file for details."
    wScript.Quit
End If
'-----
'--- now that the loader is finished, put the
'--- SharedMemoryOn registry key back to its original
'--- value.
'-----
If SharedMemoryRegKey = 1 Then
    WshShell.RegWrite
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLS
erver\Client\SharedMemoryOn", 1, "REG_DWORD"
End If
'-----
'--- set post-load options
'-----
wScript.Echo FormatDateTime(Now,0) & " ==>

```

```

Setting database options after load..."
    Call WriteBuildLog("Set post-load database options
(DBOPT2)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
UtilityDirectory & "dbopt2.sql -o" & LogDirectory &
"Database_Options_2.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"Database_Options_2.log")
    If rc <> 0 Then
        Call WriteBuildLog("Set post-load
database options (Database_Options_2)","Step failed! - Check
Database_Options_2.log")
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==>
Post-load database options set."
    Call WriteBuildLog("Post-load database options
set", "")
    wScript.Echo ""
    Call WriteBuildLog("Database load and index
creation complete", "")
    wScript.Echo FormatDateTime(Now,0) & " ==>
Database load and index creation complete."
    wScript.Echo ""
    '-----
    '--- now parse the index creation logs
    '--- to see if there were any errors
    '--- there.
    '-----
    For i = 5 To 15
        rc = CheckSQLOutput(LogDirectory & LogFileArray(i))
        If rc <> 0 Then
            wScript.Quit
        End If
    Next
    wScript.Echo FormatDateTime(Now,0) & " ==>
Calculating initial database space usage..."
    Call WriteBuildLog("Calculate TPC-C initial
database space usage", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
ACIDDirectory & "space\scripts\spused.sql -o" & LogDirectory
& "spused.ver")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
ACIDDirectory & "space\scripts\splog.sql -o" & LogDirectory &
"splog.ver")

```

```

Do While oExec.Status = 0
    wScript.Sleep 100
Loop
Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -i" &
ACIDDirectory & "space\scripts\spfiles.sql -o" & LogDirectory
& "spfiles.ver")
Do While oExec.Status = 0
    wScript.Sleep 100
Loop
wScript.Echo FormatDateTime(Now,0) & " ==>
Initial database space usage calculated."
Call WriteBuildLog("TPC-C initial database space
usage calculated", "")
wScript.Echo ""
'-----
'--- now that the loader is finished
'--- check the .err files and if they
'--- are of zero length, delete them.
'-----
Set fsErr = CreateObject("Scripting.FileSystemObject")
Set fErr = fsErr.GetFolder(LogDirectory)
Set fcErr = fErr.Files
For Each fl In fcErr
    If fl.Type = "ERR File" Then
        If fl.Size = 0 Then
            fl.Delete
        End If
    End If
Next
Set fcErr = Nothing
Set fErr = Nothing
Set fsErr = Nothing
End If
If (BuildOption = "full" Or BuildOption = "objectsfull" Or
BuildOption = "bulkloadfull" Or BuildOption = "backup") Then
    wScript.Echo FormatDateTime(Now,0) & " ==> Creating
backup device(s)..."
    Call WriteBuildLog("Creating backup device(s)", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P"
& SQLPassword & " -S" & ServerName & " -e -i" &
DBDirectory & "backupdev.sql -o" & LogDirectory &
"backupdev.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "backupdev.log")
    If rc <> 0 Then
        Call WriteBuildLog("Creating backup device(s)", "Step
failed! - Check backupdev.log")
        wScript.Echo FormatDateTime(Now,0) & " ==>
Backup device(s) creation failed! Check backupdev.log."
        wScript.Quit
    End If
End If

```

```

wScript.Echo FormatDateTime(Now,0) & " ==> Backup
device(s) created."
    Call WriteBuildLog("Backup device(s) created", "")
    wScript.Echo ""
    wScript.Echo FormatDateTime(Now,0) & " ==> Backing up
TPC-C database..."
    Call WriteBuildLog("Backing up TPC-C database", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID & " -P"
& SQLPassword & " -S" & ServerName & " -e -i" &
DBDirectory & "backup.sql -o" & LogDirectory & "backup.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory & "backup.log")
    If rc <> 0 Then
        Call WriteBuildLog("Backing up TPC-C database", "Step
failed! - Check backup.log")
        wScript.Echo FormatDateTime(Now,0) & " ==>
Database backup failed! Check backup.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==> Database
backup complete."
    wScript.Echo ""
    Call WriteBuildLog("Database backup complete", "")
End If
If (BuildOption = "full" Or BuildOption = "objectsfull" Or
BuildOption = "bulkloadfull") Then
    '-----
    '--- run a data load verification script
    '-----
    wScript.Echo FormatDateTime(Now,0) & " ==>
Verify initial TPC-C database load..."
    Call WriteBuildLog("Verify TPC-C initial database
load", "")
    Set oExec = WshShell.Exec("osql -U" & SQLUserID
& " -P" & SQLPassword & " -S" & ServerName & " -e -w300 -
i" & UtilityDirectory & "VerifyTPCCLoad.sql -o" &
LogDirectory & "VerifyTPCCLoad.log")
    Do While oExec.Status = 0
        wScript.Sleep 100
    Loop
    rc = CheckSQLOutput(LogDirectory &
"VerifyTPCCLoad.log")
    If rc <> 0 Then
        Call WriteBuildLog("Verify TPC-C initial database
load", "Step failed! - Check VerifyTPCCLoad.log")
        wScript.Echo FormatDateTime(Now,0) & " ==>
TPC-C database verification failed! Check
VerifyTPCCLoad.log."
        wScript.Quit
    End If
    wScript.Echo FormatDateTime(Now,0) & " ==>
TPC-C initial database load verified."

```

```

    Call WriteBuildLog("TPC-C initial database load
verified", "")
    wScript.Echo ""
End If
'-----
'--- display banner message
'-----
wScript.Echo
"*****"
wScript.Echo "*"
wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & " "*"
wScript.Echo "*"
wScript.Echo "*" Database Setup Complete
*"
wScript.Echo "*"
wScript.Echo
"*****"
"*****"
wScript.Quit
'-----
'--- ShowUsage
'-----
Function ShowUsage()
    wScript.Echo
    "*****"
    "*****"
    wScript.Echo "*"
    "*"
    wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & " "*"
    wScript.Echo "*"
    "*"
    wScript.Echo "*" Usage:
    "*"
    wScript.Echo "*" Optionally, you can pass the following
positional arguments to SETUP "*"
    wScript.Echo "*" /S:<Server Name> (can be "" for
local host) "*"
    wScript.Echo "*" /U:<SQL Server User ID>
(recommended you use sa) "*"
    wScript.Echo "*" /P:<SQL Sever Account Password>
    "*"
    wScript.Echo "*" (enter BLANK if you do not have a
password defined) "*"
    wScript.Echo "*" /W:<Number of Warehouses to Build>
    "*"
    wScript.Echo "*" /B:<Build Option>
    "*"
    wScript.Echo "*"
    [full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup] "*"
    wScript.Echo "*" /D:<Database Type>
    "*"

```



```

TempResponse = "bulkload"
Flag = 1
Case "bulkloadfull"
TempResponse = "bulkloadfull"
Flag = 1
Case "backup"
TempResponse = "backup"
Flag = 1
Case Else
rc = MsgBox("Invalid Database
Build Option.", 21)

If rc = 2 Then
wScript.Echo ""
wScript.Echo "TPC-C Setup
cancelled by user."

wScript.Quit
End If
Flag = 0
TempResponse = InputBox("Build
Option:" & Chr(13) &
"(full,builddb,objects,objectsfull,bulkload,bulkloadfull,backup)",
,"full")

TempResponse =
LCase(TempResponse)
End Select
Loop

Case
"DatabaseType"
TempResponse = InputBox("Database Type:" &
Chr(13) & "(normal or scale_down)", "TPC-C Setup (V" &
Kit_Version & ")", "normal")
TempResponse = LCase(TempResponse)
'--- set flag
Flag = 0
Do While Flag = 0
Select Case TempResponse
Case "normal"
TempResponse = "0"
Flag = 1
Case "scale_down"
TempResponse = "1"
Flag = 1
Case Else
rc = MsgBox("Invalid Database
Type.", 21)

If rc = 2 Then
wScript.Echo ""
wScript.Echo "TPC-C Setup
cancelled by user."

wScript.Quit
End If
Flag = 0
TempResponse =
InputBox("Database Type:" & Chr(13) & "(normal or

```

```

scale_down)", , "normal")
TempResponse =
LCase(TempResponse)
End Select
Loop

Case
"UnattendedBuild"
TempResponse = InputBox("Unattended Build?:"
& Chr(13) & "(true or false)", "TPC-C Setup (V" & Kit_Version
& ")", "false")
TempResponse = LCase(TempResponse)
'--- set flag
Flag = 0
Do While TempResponse = ""
rc = MsgBox("You must enter true or false
for Unattended Build.", 21)

If rc = 2 Then
wScript.Echo ""
wScript.Echo "TPC-C Setup cancelled
by user."

wScript.Quit
End If
TempResponse = InputBox("Unattended
Build?:" & Chr(13) & "(true or false)", "TPC-C Setup (V" &
Kit_Version & ")", "false")
TempResponse = LCase(TempResponse)
Loop

End Select
GetUserInput = TempResponse
End Function
'-----
'--- end function
'-----
'--- subroutine to write BuildLog data
'-----
Sub WriteBuildLog(StepMessage, ErrorMessage)
Set StepLog = fs.OpenTextFile(BuildStepLogFile, 8,
true)
If LEN(ErrorMessage) > 0 Then
msg = FormatDateTime(Now,0) & " ==>
" & StepMessage & " : " & ErrorMessage
Else
msg = FormatDateTime(Now,0) & " ==>
" & StepMessage
End If
StepLog.WriteLine (msg)
StepLog.close
End Sub
'-----
'--- end sub
'-----
'--- function to check for registry key existence

```

```

'-----
Function CheckRegKey(RegStr)
On Error Resume Next
WshShell.RegRead RegStr
If Err Then
CheckRegKey = False
Else
CheckRegKey = True
End If
On Error Goto 0
End Function
'-----
'--- end function
'-----

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;
}

```

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;
    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];

```

```

    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;

```

```

HDBC // for ORDERS
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;

TPCCLDR_ARGS *aptr, args;

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

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

```

```

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

printf("\n*****\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");

// 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
            long 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 bcp[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("idxitmc1");

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

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

            strcpy(err_log_path, aptr->log_path);
            strcat(err_log_path, "item.err");
            rc = bcp_init(i_hdbc1, name, NULL, err_log_path ,
DB_IN);
            if (rc != SUCCEED)

```

```

        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 != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);
    rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(i_hdbc1);

    time_start = (TimeNow() / MILLI);

    item_rows_loaded = 0;

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

        MakeAlphaStringPadded(14, 24,
I_NAME_LEN, i_name);

        i_price = ((float) RandomNumber(100L,
10000L))/100.0;

        MakeOriginalAlphaString(26, 50,
I_DATA_LEN, i_data, 10);

        rc = bcp_sendrow(i_hdbc1);

```

```

        if (rc != SUCCEEDED)
            HandleErrorDBC(i_hdbc1);

        item_rows_loaded++;
        CheckForCommit(i_hdbc1, i_hstmt1,
item_rows_loaded, "item", &time_start);
    }

    rcint = bcp_done(i_hdbc1);
    if (rcint < 0)
        HandleErrorDBC(i_hdbc1);

    printf("Finished loading item table.\n");

    SQLFreeStmt(i_hstmt1, SQL_DROP);
    SQLDisconnect(i_hdbc1);
    SQLFreeConnect(i_hdbc1);

    // if build index after load
    if ((aptr->build_index == 1) && (aptr->index_order
== 0))
        BuildIndex("idxitmc1");
    }

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

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0,
W_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_street_1, 0,

```

```

ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) w_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);

    time_start = (TimeNow() / MILLI);

    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("idxwarcl");

    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;
    long       d_next_o_id;
    long       time_start;
    long       w_id;
    RETCODE    rc;
    DBINT     rcint;
    char       bcphint[128];
    char       err_log_path[256];

    // Seed with unique number
    seed(4);

    printf("Loading district table...\n");

    // build index before load
    if ((aptr->build_index == 1) && (aptr->index_order
== 1))
        BuildIndex("idxdiscl");

    InitString(d_name, D_NAME_LEN+1);
    InitAddress(d_street_1, d_street_2, d_city, d_state,

```

```

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

    strcpy(err_log_path, aptr->log_path);
    strcat(err_log_path, "district.err");
    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 (d_w_id,
d_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
10));
        rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(w_hdbc1);
    }

    i = 0;
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) &d_tax, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) d_name, 0,
D_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) d_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) d_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)

```

```

        HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_city, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0,
STATE_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0,
ZIP_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        d_ytd = 30000.0;

        d_next_o_id = orders_per_district+1;

        time_start = (TimeNow() / MILLI);

        for (w_id = apr->starting_warehouse; w_id <= apr-
>num_warehouses; w_id++)
        {
            d_w_id = w_id;

            for (d_id = 1; d_id <=
DISTRICT_PER_WAREHOUSE; d_id++)
            {

                MakeAlphaStringPadded(6,10,D_NAME_LEN,
d_name);

                MakeAddress(d_street_1,
d_street_2, d_city, d_state, d_zip);

                d_tax = ((float)
RandomNumber(0L,2000L))/10000.00;

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

                    HandleErrorDBC(w_hdbc1);

                district_rows_loaded++;
                CheckForCommit(w_hdbc1,
w_hstmt1, district_rows_loaded, "district", &time_start);
            }
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
            HandleErrorDBC(w_hdbc1);

```

```

        printf("Finished loading district table.\n");

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order
== 0))

            BuildIndex("idxdisc1");

        return;
    }

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

    void Stock()
    {
        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 bcpint[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("idxstk1");

        sprintf(name, "%s..%s", apr->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 != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order
== 1))
        {
            sprintf(bcpint, "tablock, order (s_i_id,
s_w_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses
* 100000));

            rc = bcp_control(w_hdbc1, BCPHINTS,
(void*) bcpint);
            if (rc != SUCCEEDED)
                HandleErrorDBC(w_hdbc1);
        }

        i = 0;
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0,
SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEEDED)
            HandleErrorDBC(w_hdbc1);
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0,

```

```

S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);
    rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0,
S_DIST_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(w_hdbc1);

    s_ytd = s_order_cnt = s_remote_cnt = 0;

    time_start = (TimeNow() / MILLI);

    printf("...Loading stock table\n");

    for (s_i_id=1; s_i_id <= max_items; s_i_id++)
    {

        for (s_w_id = (long)aptr-
>starting_warehouse; s_w_id <= aptr->num_warehouses;
s_w_id++)
        {

            s_quantity =
(short)RandomNumber(10L,100L);
            len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);

```

```

len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
len =
MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

len =
MakeOriginalAlphaString(26,50, S_DATA_LEN, s_data,10);

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

    HandleErrorDBC(w_hdbc1);

    stock_rows_loaded++;

    CheckForCommit_Big(w_hdbc1, w_hstmt1,
stock_rows_loaded, "stock", &time_start);
    }
}

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
bcp[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", aprtr->database, "customer");

strcpy(err_log_path_cust, aprtr->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", aprtr->database, "history");

rc = bcp_init(c_hdbc2, name, NULL,
"logs\\history.err", DB_IN);
strcpy(err_log_path_hist, aprtr->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,
(LPCTSTR) 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,
(LPCTSTR) 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 != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
}

```

```

    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0,
LAST_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0,
FIRST_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0,
SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0,
ADDRESS_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL,
0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
}

```

```

    rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0,
C_SINCE_LEN, NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0,
MIDDLE_NAME_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, C_DATA_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
        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 != SUCCEEDED)
            HandleErrorDBC(c_hdbc1);
    }
}

```



```

        customer_rows_loaded++;
        CheckForCommit(c_hdbc1, c_hstmt1,
customer_rows_loaded, "customer", &customer_time_start-
>time_start);
    }
}

```

```

//=====
//

```

```

// Function : LoadHistoryTable
//

```

```

//=====
//

```

```

void LoadHistoryTable(LOADER_TIME_STRUCT
*history_time_start)
{

```

```

    long          i;
    long          c_id;
    short         c_d_id;
    long          c_w_id;
    double        h_amount;
    char          h_data[H_DATA_LEN+1];
    char          h_date[H_DATE_LEN+1];

```

```

    RETCODE      rc;

```

```

    i = 0;
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0,
H_DATE_LEN, NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

```

```

    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN,
NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(c_hdbc2);

```

```

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

```

```

        c_id = customer_buf[i].c_id;
        c_d_id = customer_buf[i].c_d_id;
        c_w_id = customer_buf[i].c_w_id;
        h_amount = customer_buf[i].h_amount;
        strcpy(h_data, customer_buf[i].h_data);

```

```

        FormatDate(&h_date);

```

```

        // send to server
        rc = bcp_sendrow(c_hdbc2);
        if (rc != SUCCEED)

```

```

            HandleErrorDBC(c_hdbc2);

```

```

            history_rows_loaded++;
            CheckForCommit(c_hdbc2, c_hstmt2,

```

```

history_rows_loaded, "history", &history_time_start-
>time_start);
        }
    }

```

```

//=====
//

```

```

// Function : LoadOrders
//

```

```

//=====
//

```

```

void LoadOrders()
{

```

```

    LOADER_TIME_STRUCT  orders_time_start;
    LOADER_TIME_STRUCT  new_order_time_start;
    LOADER_TIME_STRUCT  order_line_time_start;
    long                w_id;
    short               d_id;
    DWORD               dwThreadID[MAX_ORDER_THREADS];
    HANDLE              hThread[MAX_ORDER_THREADS];

```

```

    char                name[20];

```

```

    RETCODE

```

```

        rc;

```

```

    char                bcphint[128];
    char                err_log_path_ord[256];
    char                err_log_path_nord[256];
    char                err_log_path_ordl[256];

```

```

    // seed with unique number
    seed(6);

```

```

    printf("Loading orders...\n");

```

```

    // if build index before load...

```

```

    if ((aptr->build_index == 1) && (aptr->index_order
== 1))
    {

```

```

        BuildIndex("idxordcl");
        BuildIndex("idxnodcl");
        BuildIndex("idxodcl");
    }

```

```

    // 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 * 30000));
        rc = bcp_control(o_hdbc1, BCPHINTS,

```

```

(void*) bcphint);
        if (rc != SUCCEED)
            HandleErrorDBC(o_hdbc1);
    }

```

```

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

```

```

    rc = bcp_init(o_hdbc2, name, NULL,
"logs\\neword.err", DB_IN);

```

```

strcpy(err_log_path_nord,aptr->log_path);
strcat(err_log_path_nord,"neword.err");
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 * 30000));
    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,

        (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,

        (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 != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0,
SQL_VARLEN_DATA, NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc1);

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

```

```

        o_id    = orders_buf[i].o_id;
        o_d_id  = orders_buf[i].o_d_id;
        o_w_id  = orders_buf[i].o_w_id;
        o_c_id  = orders_buf[i].o_c_id;
        o_carrier_id = orders_buf[i].o_carrier_id;
        o_ol_cnt = orders_buf[i].o_ol_cnt;
        o_all_local = orders_buf[i].o_all_local;

        FormatDate(&o_entry_d);

        // send data to server
        rc = bcp_sendrow(o_hdbc1);
        if (rc != SUCCEEDED)
            HandleErrorDBC(o_hdbc1);

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

    if ((o_w_id == apr->num_warehouses) && (o_d_id
== 10))
    {
        rcint = bcp_done(o_hdbc1);

        if (rcint < 0)
            HandleErrorDBC(o_hdbc1);

        SQLFreeStmt(o_hstmt1, SQL_DROP);
        SQLDisconnect(o_hdbc1);
        SQLFreeConnect(o_hdbc1);

        // if build index after load...
        if ((apr->build_index == 1) && (apr-
>index_order == 0))
            BuildIndex("idxordcl");

        // build non-clustered index
        if (apr->build_index == 1)
            BuildIndex("idxordnc");
    }

}

//=====
//
// Function : LoadNewOrderTable
//
//=====

```

```

void LoadNewOrderTable(LOADER_TIME_STRUCT
*new_order_time_start)
{
    long   long           i;
    long   o_id;
    short  o_d_id;
    long   o_w_id;
    RETCODE rc;
    DBINT  rcint;

    // Bind NEW-ORDER data
    i = 0;
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEEDED)
        HandleErrorDBC(o_hdbc2);

    for (i = first_new_order; i < last_new_order; i++)
    {
        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 != SUCCEEDED)
            HandleErrorDBC(o_hdbc2);

        new_order_rows_loaded++;

        CheckForCommit_Big(o_hdbc2, o_hstmt2,
new_order_rows_loaded, "new_order", &new_order_time_start-
>time_start);
    }

    if ((o_w_id == apr->num_warehouses) && (o_d_id
== 10))
    {
        rcint = bcp_done(o_hdbc2);

        if (rcint < 0)
            HandleErrorDBC(o_hdbc2);

        SQLFreeStmt(o_hstmt2, SQL_DROP);
    }
}

```

```

SQLDisconnect(o_hdbc2);
SQLFreeConnect(o_hdbc2);

// if build index after load...
if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
    BuildIndex("idxnodel");
}
}

//=====
//
// Function : LoadOrderLineTable
//
//=====

void LoadOrderLineTable(LOADER_TIME_STRUCT
*order_line_time_start)
{
    long i;
    long j;
    long o_id;
    short o_d_id;
    long o_w_id;
    double ol;
    long ol_i_id;
    long ol_supply_w_id;
    short ol_quantity;
    double ol_amount;
    char ol_dist_info[DIST_INFO_LEN+1];
    char ol_delivery_d[OL_DELIVERY_D_LEN+1];
    RETCODE rc;
    DBINT rcint;

    // bind ORDER-LINE data
    i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

```

```

rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0,
OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0,
SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id,
0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0,
SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0,
DIST_INFO_LEN, NULL, 0, 0, ++i);
    if (rc != SUCCEED)
        HandleErrorDBC(o_hdbc3);

    for (i = 0; i < orders_per_district; i++)
    {
        o_id = orders_buf[i].o_id;
        o_d_id = orders_buf[i].o_d_id;
        o_w_id = orders_buf[i].o_w_id;

        for (j=0; j < orders_buf[i].o_ol_cnt; j++)
        {
            ol =
orders_buf[i].o_ol[j].ol;
            ol_i_id =
orders_buf[i].o_ol[j].ol_i_id;
            ol_supply_w_id =
orders_buf[i].o_ol[j].ol_supply_w_id;
            ol_quantity =
orders_buf[i].o_ol[j].ol_quantity;
            ol_amount =
orders_buf[i].o_ol[j].ol_amount;

            strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery
_d);

            strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

            rc = bcp_sendrow(o_hdbc3);
            if (rc != SUCCEED)

```

```

HandleErrorDBC(o_hdbc3);
                order_line_rows_loaded++;
        }
        CheckForCommit_Big(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start-
>time_start);
    }
}

if ((o_w_id == aptr->num_warehouses) && (o_d_id
== 10))
{
    rcint = bcp_done(o_hdbc3);
    if (rcint < 0)
        HandleErrorDBC(o_hdbc3);

    SQLFreeStmt(o_hstmt3, SQL_DROP);
    SQLDisconnect(o_hdbc3);
    SQLFreeConnect(o_hdbc3);

    // if build index after load...
    if ((aptr->build_index == 1) && (aptr-
>index_order == 0))
        BuildIndex("idxodlcl");
}
}

//=====
//
// Function : GetPermutation
//
//=====

void GetPermutation(int perm[], int n)
{
    int i, r, t;

    for (i=1; i<=n; i++)
        perm[i] = i;

    for (i=1; i<=n; i++)
    {
        r = RandomNumber(i,n);
        t = perm[i];
        perm[i] = perm[r];
        perm[r] = t;
    }
}

```

```

=====
//
// Function : CheckForCommit
//
=====
void CheckForCommit(HDBC hdbc,
                   HSTMT
hstmt,
                   long
rows_loaded,
                   char
*table_name,
                   long *time_start)
{
    long    time_end, time_diff;

    if ( !(rows_loaded % apr->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",
              apr->batch,
              table_name,
              time_diff,
              rows_loaded,
              (float) apr->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, apr->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",
              apr->batch,
              table_name,
              time_diff,
              rows_loaded,
              (float) apr->batch /
(time_diff ? time_diff : 1L));

        *time_start = time_end;
    }

    return;
}
=====
//
// Function : OpenConnections
//
=====
void OpenConnections()
{
    RETCODE    rc;

    char
char                szDriverString[300];
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" ,

            apr->server,

            apr->user,

            apr->password,

            apr->database );

    rc = SQLSetConnectOption ( i_hdbc1,
SQL_PACKET_SIZE, apr->pack_size);
    if (rc != SUCCEED)
        HandleErrorDBC(i_hdbc1);

    rc = SQLDriverConnect ( i_hdbc1,
NULL,

(SQLCHAR*)&szDriverString[0] ,

SQL_NTS,

(SQLCHAR*)&szDriverStringOut[0],

sizeof(szDriverStringOut),

&cbDriverStringOut,

```

```

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

rc = SQLDriverConnect ( w_hdbc1,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );

if ( (rc != SUCCEEDED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(w_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 3
sprintf( szDriverString , "DRIVER={SQL

```

```

Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

    aptr->server,

    aptr->user,

    aptr->password,

    aptr->database );

rc = SQLSetConnectOption (c_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc1);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 4
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

    aptr->server,

    aptr->user,

    aptr->password,

    aptr->database );

rc = SQLSetConnectOption (c_hdbc2,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(c_hdbc2);

```

```

rc = SQLDriverConnect ( c_hdbc2,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

    SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEEDED) &&
    (rc != SQL_SUCCESS_WITH_INFO) )
{
    HandleErrorDBC(c_hdbc2);
    printf("TPC-C Loader aborted!\n");
    exit(9);
}

// Connection 5
sprintf( szDriverString , "DRIVER={SQL
Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,

    aptr->server,

    aptr->user,

    aptr->password,

    aptr->database );

rc = SQLSetConnectOption (o_hdbc1,
SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEEDED)
    HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,

    NULL,

    (SQLCHAR*)&szDriverString[0] ,

    SQL_NTS,

    (SQLCHAR*)&szDriverStringOut[0],

    sizeof(szDriverStringOut),

    &cbDriverStringOut,

```

```

        SQL_DRIVER_NOPROMPT );
if ( (rc != 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 )
    {
        printf( szLastError , "%s" , Msg );

        _strtime(timebuf);
        _strdate(datebuf);

        printf( "[%s : %s] %s\n====SQLState:
%s\n" , datebuf, timebuf, szLastError, SqlState);

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"tpccldr.err");
        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");
fp1 = fopen(err_log_path,"a+");
if (fp1 == NULL)
    printf("ERROR: Unable to
open errorlog file.\n");
else
    {
        fprintf(fp1, "[%s : %s]
%s\nSQLState: %s\n", datebuf, timebuf, szLastError, SqlState);
        fclose(fp1);
    }
    i++;
}

//=====
//
// Function : FormatDate
//
//=====

void FormatDate ( char* szTimeCOutput )
{
    struct tm when;
    time_t now;

    time( &now );
    when = *localtime( &now );

    mktime( &when );

    // odbc datetime format
    strftime( szTimeCOutput , 30 , "%Y-%m-%d
%H:%M:%S.000" , &when );

    return;
}

```

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

```

```

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

```

```

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

```

```

    return rand_num;
}

```

runsqlcfg.vbs

```

'-----
'--- FILE:   RESTORE.VBS
'---        Microsoft TPC-C Kit Ver. 4.68
'---        Copyright Microsoft, 2001, 2002, 2006
'---        All Rights Reserved
'---
'--- PURPOSE: This module executes a database restore
'---
'-----
'--- set the kit version variable for later display
'-----
Kit_Version = " 4.68"
'-----
'--- open an windows scripting object
'-----
set WshShell = CreateObject("WScript.Shell")
'-----
'--- set up windows scripting argument collection
'-----
Set objArgs = wScript.Arguments
'-----
'--- display a banner message
'-----
wScript.Echo
"*****"
"*****"
wScript.Echo "*"
wScript.Echo "*" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & "          *"
wScript.Echo "*"

```

```

wScript.Echo "*" Configure SQL Server
*"
wScript.Echo "*"
wScript.Echo
"*****"
"*****"
'-----
'--- define function to check for any error messages
'-----
Function CheckSQLOutput(SQL_Out)
    ErrorFlag = 0
    Set SQL_fso =
CreateObject("Scripting.FileSystemObject")
    Set SQL_Out_File =
SQL_fso.OpenTextFile(SQL_Out,1)
    Do While SQL_Out_File.AtEndOfStream <> True
        SQL_Line = SQL_Out_File.ReadLine
        'first check to see if the output contains a
message about the login password
        If InStr(SQL_Line, "Login failed") Then
            'display the messages and get
            out of here
            ErrorFlag = 1
            Wscript.Echo "The login for
userid 'sa' failed."
            Wscript.Echo "Please restart
SETUP with the correct password."
            End If
        Loop
        SQL_Out_File.Close
        CheckSQLOutput = ErrorFlag
    End Function
'-----
'--- end function
'-----
'--- open a file system object
'-----
Set fs = CreateObject("Scripting.FileSystemObject")
'-----
'--- grab the current directory value
'-----
SetupDirectory = WshShell.CurrentDirectory & "\"
'-----
'--- now calculate the other directories
'-----
ScriptDirectory = SetupDirectory & "SCRIPTS\"
LogDirectory = SetupDirectory & "LOGS\"
'-----
'--- check to see if the user passed in the server name and sa
password
'-----
Select Case objArgs.Length
    Case 0

```

```

'-----
'--- the user did not pass us anything
'--- grab the Computer Name from
Windows
'-----
ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%
")
'-----
'--- prompt the user to confirm the server
name
'-----
ServerName = InputBox("Enter your
server name",Test_Name,ServerName)
Do While ServerName = ""
    rc = MsgBox ("You must enter
a valid server name.",21)
    If rc = 2 Then
        wScript.Echo ""
        wScript.Echo "TPC-
C Setup cancelled by user."
        wScript.Quit
    End If
    ServerName =
WshShell.ExpandEnvironmentStrings("%COMPUTERNAME%
")
    ServerName = InputBox("Enter
your server name","Database Server Name",ServerName)
    Loop
'-----
'--- prompt the user for the sa password
'-----
saPassword = InputBox("Enter the 'sa'
password",Test_Name)
Case 1
'-----
'--- the user passed 1 argument, so assume
it is the server name
'-----
ServerName = objArgs(0)
'-----
'--- prompt the user for the sa password
'-----
saPassword = InputBox("Enter the 'sa'
password",Test_Name)
Case 2
'-----
'--- the user passed 2 arguments, so try to
use them
'-----
ServerName = objArgs(0)
saPassword = objArgs(1)

```

```

End Select
'-----
'--- now that we have all the variables filled in, let's get to work
'-----
If fs.FileExists(LogDirectory & "runsqlcfg.log") Then
    fs.DeleteFile LogDirectory & "runsqlcfg.log"
End If
'-----
'--- configure SQL Server
'-----
wScript.Echo " "
wScript.Echo "Configuring Microsoft SQL Server installation..."
Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -
S" & ServerName & " -e -i" & ScriptDirectory &
"utility/runsqlcfg.sql -o" & LogDirectory & "runsqlcfg.log")
Do While oExec.Status = 0
    WScript.Sleep 100
Loop
rc = CheckSQLOutput(LogDirectory & "runsqlcfg.log")
If rc <> 0 Then
    WScript.Quit
End If
wScript.Echo " "
wScript.Echo "SQL Server Configuration Complete."
'-----
'--- shutdown SQL Server
'-----
wScript.Echo " "
wScript.Echo "Shutting down SQL Server..."
Set oExec = WshShell.Exec("osql -Usa -P" & saPassword & " -
S" & ServerName & " -e -i" & ScriptDirectory &
"utility/sqlshutdown.sql")
wScript.Echo " "
wScript.Echo "Waiting for SQL Server to shutdown..."
Set oExec = WshShell.Exec("..\tools\sleep\sleep.exe 20")
Do While oExec.Status = 0
    WScript.Sleep 100
Loop
'-----
'--- Restarting SQL Server
'-----
wScript.Echo " "
wScript.Echo "Restarting SQL Server..."
wScript.Echo " "
CMD_String = "start sqlservr.exe -c -t3502"
oExec = WshShell.Run(CMD_String, 2, false)
wScript.Echo ""
wScript.Echo ""
*****
*****
wScript.Echo ""
wScript.Echo "" Microsoft TPC-C Benchmark Kit Ver." &
Kit_Version & "
wScript.Echo ""

```

```

wScript.Echo "" SQL Server configuration complete.
*"
wScript.Echo ""
wScript.Echo
*****
*****

```

strings.c

```

//      File:                STRINGS.C
//                               Microsoft TPC-C
Kit Ver. 4.51
//                               Copyright Microsoft,
1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//      Purpose:   Source file for database loader string
functions

// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//=====
//
// Function name: MakeAddress
//
//=====

void MakeAddress(char *street_1,
                 char *street_2,
                 char *city,
                 char *state,
                 char *zip)
{
#ifdef DEBUG
    printf("[%d]DBG: Entering MakeAddress()\n", (int)
GetCurrentThreadId());
#endif

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString (9, 9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%d]DBG: MakeAddress: street_1: %s, street_2: %s,
city: %s, state: %s, zip: %s\n",

```

```

(int) GetCurrentThreadId(),
street_1, street_2, city, state, zip);
#endif

    return;
}

//=====
//
// Function name: LastName
//
//=====

void LastName(int num,
             char *name)
{
    static char *n[] =
    {
        "BAR", "OUGHT", "ABLE", "PRI",
"PRES",
        "ESE", "ANTI", "CALLY", "ATION",
"EING"
    };

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

    if ((num >= 0) && (num < 1000))
    {
        strcpy(name, n[(num/100)%10]);
        strcat(name, n[(num/10)%10]);
        strcat(name, n[(num/1)%10]);

        if (strlen(name) < LAST_NAME_LEN)
        {
            PaddString(LAST_NAME_LEN, name);
        }
    }
    else
    {
        printf("\nError in LastName()... num
<%d> out of range (0,999)\n", num);
        exit(-1);
    }
}

```

```

#ifdef DEBUG
    printf("[%d]DBG: LastName: num = [%d] ==>
[%d][%d][%d]\n",
        (int) GetCurrentThreadId(),
        num, num/100, (num/10)%10, num%10);
    printf("[%d]DBG: LastName: String = %s\n", (int)
GetCurrentThreadId(), name);
#endif

    return;
}

//=====
//
// Function name: MakeAlphaString
//
//=====

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z,
and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random
alphanumeric
//(respectively, numeric) characters of a random length of
minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9. The
only other
//requirement is that the character set used "must be able to
represent a minimum
//of 128 different characters". We are using 8-bit chars, so this is
a non issue.
//It is completely unreasonable to stuff non-printing chars into
the text fields.
//--CLevine 08/13/96

int MakeAlphaString( int x, int y, int z, char *str)
{
    int len;
    int i;
    char cc = 'a';
    static char chArray[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefg
hijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

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

```

```

#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[] =
"0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefg
hijklmnopqrstuvwxyz";
    static int chArrayMax = 61;

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

    len= RandomNumber(minLen, maxLen);

    for (i=0; i<len; i++)
        str[i] =
chArray[RandomNumber(0,chArrayMax)];
    if (len < padLen)
        memset(str+len, ' ', padLen - len);
    str[padLen] = 0;
    return padLen;
}

//=====
//
// Function name: MakeOriginalAlphaString
//
//=====

int MakeOriginalAlphaString(int x,

    int y,

    int z,

```

```

        char *str,

        int percent)
{
    int len;
    int val;
    int start;

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

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

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

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

    return len;
}

//=====
//
// Function name: MakeNumberString

```

```

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

    //MakeNumberString is always called
    MakeZipNumberString(16, 16, 16, string)

    memset(str, '0', 16);
    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    itoa(RandomNumber(0, 99999999), tmp, 10);
    memcpy(str+8, tmp, strlen(tmp));

    str[16] = 0;

    return 16;
}

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

    //MakeZipNumberString is always called
    MakeZipNumberString(9, 9, 9, string)

    strcpy(str, "000011111");

    itoa(RandomNumber(0, 9999), tmp, 10);
    memcpy(str, tmp, strlen(tmp));

    return 9;
}

//=====
//=====
//
// Function name: InitString
//
//=====
//=====
void InitString(char *str, int len)

```

```

{
#ifdef DEBUG
    printf("[%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;
}

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; // set if loading all
    BOOL table_item; // set if loading
    ITEM table specifically
    BOOL table_warehouse; // set if loading WAREHOUSE,
    DISTRICT, and STOCK
    BOOL table_customer; // set if loading
    CUSTOMER and HISTORY
    BOOL table_orders; // set if loading
    NEW-ORDER, ORDERS, ORDER-LINE
    long num_warehouses;
    long batch;
    long verbose;

    long pack_size;
    char *loader_res_file;
    char *log_path;
    char *synch_servername;
    long

```

```

case_sensitivity;
long starting_warehouse;
long build_index;
long index_order;
long scale_down;
char *index_script_path;
} TPCCLDR_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();

```

Appendix C: Tunable Parameters

Database 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)

```

```
-----
2008-08-28 15:43:56.727
```

```

(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 ''
-----
2008-08-28 15:43:56.977

(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
16777215 16777215
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          0          0

```

```

disallow results from triggers      0          1          0
0
fill factor (%)                     0          100         0          0
ft crawl bandwidth (max)             0          32767      100
100
ft crawl bandwidth (min)             0          32767      0
0
ft notify bandwidth (max)            0          32767      100
100
ft notify bandwidth (min)            0          32767      0
0
in-doubt xact resolution              0          2          0          0
index create memory (KB)             704 2147483647
0          0
lightweight pooling                   0          1          1          1
locks                                5000 2147483647 0          0
max degree of parallelism             0          64          1
1
max full-text crawl range             0          256         4
4
max server memory (MB)               16 2147483647
253536 253536
max text repl size (B)               0 2147483647 65536
65536
max worker threads                   128 32767 1250
1250
media retention                       0          365         0          0
min memory per query (KB)            512 2147483647
1024 1024
min server memory (MB)               0 2147483647
16 16
nested triggers                      0          1          1          1
network packet size (B)              512 32767 4096
4096
Ole Automation Procedures            0          1          0
0
open objects                         0 2147483647 0
0
PH timeout (s)                       1 3600 60 60
precompute rank                      0          1          0          0
priority boost                        0          1          1          1
query governor cost limit             0 2147483647 0
0
query wait (s)                       -1 2147483647 -1
-1
recovery interval (min)              0          32767      32767
32767
remote access                         0          1          1          1
remote admin connections              0          1          0
0
remote login timeout (s)              0 2147483647 0
0
remote proc trans                     0          1          0          0
remote query timeout (s)              0 2147483647 0
0

```


IRQ 16	Broadcom BCM5709C NetXtreme II GigE		0x00003100-0x0000311F	Standard Universal PCI to USB	0x00000090-0x0000009F	Motherboard resources
IRQ 16	ATI ES1000		Host Controller OK		OK	
			0x00003200-0x0000321F	Standard Universal PCI to USB	0x000000A4-0x000000A5	Motherboard resources
IRQ 17	Broadcom BCM5709C NetXtreme II GigE		Host Controller OK		OK	
IRQ 17	Standard Universal PCI to USB Host Controller		0x00003300-0x0000331F	Standard Universal PCI to USB	0x000000A8-0x000000A9	Motherboard resources
			Host Controller OK		OK	
			0x00002000-0x000020FF	ATI ES1000 OK	0x000000AC-0x000000AD	Motherboard resources
Memory Address 0xE0000000-0xE7FFFFFF	PCI bus				OK	
Memory Address 0xE0000000-0xE7FFFFFF	ATI ES1000		0x000003B0-0x000003BB	ATI ES1000 OK	0x000000B0-0x000000B5	Motherboard resources
					OK	
			0x000003C0-0x000003DF	ATI ES1000 OK	0x000000B8-0x000000B9	Motherboard resources
					OK	
Memory Address 0xA0000-0xBFFFF	PCI bus		0x000003F8-0x000003FF	Communications Port (COM1)	0x000000BC-0x000000BD	Motherboard resources
Memory Address 0xA0000-0xBFFFF	ATI ES1000		OK		OK	
			0x00000020-0x00000021	Advanced programmable	0x00000400-0x0000043F	Motherboard resources
			interrupt controller OK		OK	
I/O Port 0x00003900-0x000039FF	PCI bus		0x000000A0-0x000000A1	Advanced programmable	0x00000440-0x0000045F	Motherboard resources
I/O Port 0x00003900-0x000039FF	LSI Adapter, SAS		interrupt controller OK		OK	
RAID-on-Chip, 8-port with 1078 -StorPort			0x00000080-0x0000008F	Direct memory access	0x000004D0-0x000004D1	Motherboard resources
			controller OK		OK	
I/O Port 0x00003C00-0x00003DFF	PCI bus		0x000000C0-0x000000DF	Direct memory access	0x00000540-0x0000055F	Motherboard resources
I/O Port 0x00003C00-0x00003DFF	QLogic Fibre		controller OK		OK	
Channel Adapter			0x00000040-0x00000043	System timer OK	0x00000560-0x0000057F	Motherboard resources
					OK	
I/O Port 0x00003A00-0x00003BFF	PCI bus		0x00000070-0x00000073	System CMOS/real time clock	0x00000580-0x000005FF	Motherboard resources
I/O Port 0x00003A00-0x00003BFF	QLogic Fibre		OK		OK	
Channel Adapter			0x00000061-0x00000061	System speaker OK	0x00000C00-0x00000CA7	Motherboard resources
					OK	
Memory Address 0xEE000000-0xF0FFFFFF	PCI bus		0x000000F0-0x000000FF	Numeric data processor	0x00000CB0-0x00000CDE	Motherboard resources
			OK		OK	
Memory Address 0xEE000000-0xF0FFFFFF	PCI		0x00000024-0x00000025	Motherboard resources	0x00000CA8-0x00000CA8	Motherboard resources
standard PCI-to-PCI bridge			OK		OK	
Memory Address 0xEE000000-0xF0FFFFFF			0x00000028-0x00000029	Motherboard resources	0x00000CAC-0x00000CAC	Motherboard resources
Broadcom BCM5708C NetXtreme II GigE			OK		OK	
			0x0000002C-0x0000002D	Motherboard resources	0x00000700-0x0000070F	Standard Dual Channel PCI
			OK		IDE Controller OK	
[DMA]			0x00000030-0x00000031	Motherboard resources	0x000001F0-0x000001F7	Primary IDE Channel OK
			OK			
Resource Device Status			0x00000034-0x00000035	Motherboard resources	0x000003F6-0x000003F6	Primary IDE Channel OK
Channel 4 Direct memory access controller OK			OK			
			0x00000038-0x00000039	Motherboard resources	0x00000170-0x00000177	Secondary IDE Channel
			OK		OK	
[Forced Hardware]			0x0000003C-0x0000003D	Motherboard resources	0x00000376-0x00000376	Secondary IDE Channel
			OK		OK	
Device PNP Device ID			0x0000004E-0x0000004F	Motherboard resources	0x00003900-0x000039FF	PCI bus OK
			OK		0x00003900-0x000039FF	LSI Adapter, SAS RAID-on-
[I/O]			0x00000050-0x00000052	Motherboard resources	Chip, 8-port with 1078 -StorPort	OK
			OK		OK	
Resource Device Status			0x00000060-0x00000060	Motherboard resources	0x00003A00-0x00003BFF	PCI bus OK
0x00000000-0x00000CF7	PCI bus OK		OK		OK	
0x00000000-0x00000CF7	Direct memory access		0x00000064-0x00000064	Motherboard resources	0x00003B00-0x00003BFF	QLogic Fibre Channel Adapter
controller OK			OK		OK	
0x00000D00-0x000033FF	PCI bus OK		0x00000074-0x00000077	Motherboard resources	0x00003C00-0x00003DFF	QLogic Fibre Channel Adapter
0x00003000-0x0000301F	Standard Universal PCI to USB		OK		0x00003C00-0x00003DFF	QLogic Fibre Channel Adapter
Host Controller OK			OK			

OK
 0x00003D00-0x00003DFF QLogic Fibre Channel Adapter
 OK
 0x00003400-0x000034FF PCI bus OK
 0x00003400-0x000034FF IBM ServeRAID-MR10M
 SAS/SATA Controller OK
 0x00003500-0x000036FF PCI bus OK
 0x00003500-0x000036FF QLogic Fibre Channel Adapter
 OK
 0x00003600-0x000036FF QLogic Fibre Channel Adapter
 OK
 0x00003700-0x000038FF PCI bus OK
 0x00003700-0x000038FF QLogic Fibre Channel Adapter
 OK
 0x00003800-0x000038FF QLogic Fibre Channel Adapter
 OK

[IRQs]

Resource	Device	Status
IRQ 9	Microsoft ACPI-Compliant System	OK
IRQ 16	Broadcom BCM5709C NetXtreme II GigE	OK
IRQ 16	ATI ES1000	OK
IRQ 17	Broadcom BCM5709C NetXtreme II GigE	OK
IRQ 17	Standard Universal PCI to USB Host Controller	OK
IRQ 23	Standard Universal PCI to USB Host Controller	OK
IRQ 23	Standard Enhanced PCI to USB Host Controller	OK
IRQ 18	Standard Universal PCI to USB Host Controller	OK
IRQ 19	Standard Universal PCI to USB Host Controller	OK
IRQ 4	Communications Port (COM1)	OK
IRQ 0	System timer	OK
IRQ 8	System CMOS/real time clock	OK
IRQ 13	Numeric data processor	OK
IRQ 14	Primary IDE Channel	OK
IRQ 46	LSI Adapter, SAS RAID-on-Chip, 8-port with 1078 - StorPort	OK
IRQ 79	QLogic Fibre Channel Adapter	OK
IRQ 83	QLogic Fibre Channel Adapter	OK
IRQ 80	QLogic Fibre Channel Adapter	OK
IRQ 84	QLogic Fibre Channel Adapter	OK
IRQ 81	Broadcom BCM5708C NetXtreme II GigE	OK
IRQ 82	Broadcom BCM5708C NetXtreme II GigE	OK
IRQ 43	IBM ServeRAID-MR10M SAS/SATA Controller	OK

IRQ 44	QLogic Fibre Channel Adapter	OK
IRQ 48	QLogic Fibre Channel Adapter	OK
IRQ 45	QLogic Fibre Channel Adapter	OK
IRQ 49	QLogic Fibre Channel Adapter	OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI ES1000	OK
0xE0000000-0xE7FFFFFF	PCI bus	OK
0xE0000000-0xE7FFFFFF	ATI ES1000	OK
0xE8000000-0xEC1FFFFFF	PCI bus	OK
0xE8000000-0xEC1FFFFFF	PCI standard PCI-to-PCI bridge	OK
0xE8000000-0xEC1FFFFFF	Broadcom BCM5709C NetXtreme II GigE	OK
0xEA000000-0xEBFFFFFF	Broadcom BCM5709C NetXtreme II GigE	OK
0xEC100000-0xEC1003FF	Standard Enhanced PCI to USB Host Controller	OK
0xEC020000-0xEC02FFFF	ATI ES1000	OK
0xFDE86000-0xFDF05FFF	Motherboard resources	OK
0xFDA00000-0xFDBFFFFFF	Motherboard resources	OK
0x0400-0x04FF	System board	OK
0xFED40000-0xFED44FFF	Motherboard resources	OK
0x100000-0xCFFFFFFF	Memory Module	OK
0xEC500000-0xEC5FFFFFF	PCI bus	OK
0xEC500000-0xEC5FFFFFF	LSI Adapter, SAS RAID-on-Chip, 8-port with 1078 - StorPort	OK
0xEC550000-0xEC553FFF	LSI Adapter, SAS RAID-on-Chip, 8-port with 1078 - StorPort	OK
0xEC600000-0xEC6FFFFFF	PCI bus	OK
0xEC640000-0xEC643FFF	QLogic Fibre Channel Adapter	OK
0xEC644000-0xEC647FFF	QLogic Fibre Channel Adapter	OK
0xEC700000-0xEC7FFFFFF	PCI bus	OK
0xEC740000-0xEC743FFF	QLogic Fibre Channel Adapter	OK
0xEC744000-0xEC747FFF	QLogic Fibre Channel Adapter	OK
0xEE000000-0xF00FFFFF	PCI bus	OK
0xEE000000-0xF00FFFFF	PCI standard PCI-to-PCI bridge	OK
0xEE000000-0xF00FFFFF	Broadcom BCM5708C NetXtreme II GigE	OK
0xF2000000-0xF40FFFFF	PCI bus	OK

0xF2000000-0xF40FFFFF	PCI standard PCI-to-PCI bridge	OK
0xF2000000-0xF40FFFFF	Broadcom BCM5708C NetXtreme II GigE	OK
0xEC200000-0xEC2FFFFFF	PCI bus	OK
0xEC200000-0xEC2FFFFFF	IBM ServeRAID-MR10M SAS/SATA Controller	OK
0xEC240000-0xEC27FFFF	IBM ServeRAID-MR10M SAS/SATA Controller	OK
0xEC300000-0xEC3FFFFFF	PCI bus	OK
0xEC340000-0xEC343FFF	QLogic Fibre Channel Adapter	OK
0xEC344000-0xEC347FFF	QLogic Fibre Channel Adapter	OK
0xEC400000-0xEC4FFFFFF	PCI bus	OK
0xEC440000-0xEC443FFF	QLogic Fibre Channel Adapter	OK
0xEC444000-0xEC447FFF	QLogic Fibre Channel Adapter	OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status
File	Version	Size	Creation Date
c:\windows\system32\imaadp32.acm	Microsoft Corporation	OK	
C:\WINDOWS\system32\IMAADP32.ACM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.00 KB (24,576 bytes)	3/25/2005 8:00 AM
c:\windows\system32\msg711.acm	Microsoft Corporation	OK	
C:\WINDOWS\system32\MSG711.ACM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	13.50 KB (13,824 bytes)	3/25/2005 8:00 AM
c:\windows\system32\msgsm32.acm	Microsoft Corporation	OK	
C:\WINDOWS\system32\MSGSM32.ACM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	34.50 KB (35,328 bytes)	3/25/2005 8:00 AM
c:\windows\system32\tsoft32.acm	DSP GROUP, INC.	OK	
C:\WINDOWS\system32\TSSOFT32.ACM	1.01		

13.50 KB (13,824 bytes) 3/25/2005 8:00 AM
 c:\windows\system32\msadp32.acm Microsoft Corporation
 OK
 C:\WINDOWS\system32\MSADP32.ACM
 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 23.50 KB (24,064 bytes) 3/25/2005 8:00 AM

[Video Codecs]

CODEC File	Manufacturer Version	Description Size	Status Creation Date
c:\windows\system32\msrle32.dll	Microsoft Corporation	OK	
C:\WINDOWS\system32\MSRLE32.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	15.50 KB (15,872 bytes)	3/25/2005 8:00 AM

c:\windows\system32\tsbyuv.dll	Microsoft Corporation	OK	
C:\WINDOWS\system32\TSBYUV.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	12.50 KB (12,800 bytes)	3/24/2005 1:34 PM

c:\windows\system32\msyuv.dll	Microsoft Corporation	OK	
C:\WINDOWS\system32\MSYUV.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	21.00 KB (21,504 bytes)	3/24/2005 1:21 PM

c:\windows\system32\iyuv_32.dll	Microsoft Corporation	OK	
C:\WINDOWS\system32\IYUV_32.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	52.50 KB (53,760 bytes)	3/24/2005 1:19 PM

c:\windows\system32\msvidc32.dll	Microsoft Corporation	OK	
C:\WINDOWS\system32\MSVIDC32.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	43.00 KB (44,032 bytes)	3/25/2005 8:00 AM

[CD-ROM]

Item	Value
Drive	D:
Description	CD-ROM Drive
Media Loaded	No
Media Type	CD-ROM
Name	HL-DT-ST RW/DVD GCC-T10N
Manufacturer	(Standard CD-ROM drives)

Status OK
 Transfer Rate Not Available
 SCSI Target ID 0
 PNP Device ID IDE\CDROMHL-DT-ST_RW/DVD_GCC-T10N
 1.00 \5&E769355&0&0.0.0

Driver c:\windows\system32\drivers\cdrom.sys
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 75.50 KB (77,312 bytes), 3/25/2005 8:00 AM)

[Sound Device]

Item	Value
------	-------

[Display]

Item	Value
Name	ATI ES1000
PNP Device ID	PCI\VEN_1002&DEV_515E&SUBSYS_03191014&REV_024&18E510B5&0&00F0
Adapter Type	ATI ES1000 (0x515E), ATI Technologies Inc. compatible
Adapter Description	ATI ES1000
Adapter RAM	16.00 MB (16,777,216 bytes)
Installed Drivers	ati2dvag.dll
Driver Version	6.14.10.6606
INF File	oem5.inf (ati2mtag_RN50 section)
Color Planes	1
Color Table Entries	4294967296
Resolution	1024 x 768 x 75 hertz
Bits/Pixel	32
Memory Address	0xE0000000-0xE7FFFFFF
I/O Port	0x00002000-0x000020FF
Memory Address	0xEC020000-0xEC02FFFF
IRQ Channel	IRQ 16
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, 2.11 MB (2,210,304 bytes), 4/5/2006 10:04 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_0624&PID_0296&MI_00&A56CC61&0&0000
Number of Function Keys	12
Driver	c:\windows\system32\drivers\hidusb.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 18.50 KB (18,944 bytes), 3/25/2005 8:00 AM)

Description	USB Human Interface Device
Name	Enhanced (101- or 102-key)
Layout	00000409
PNP Device ID	USB\VID_04B3&PID_4001&MI_00&6&2E1ACEF8&0&0000
Number of Function Keys	12
Driver	c:\windows\system32\drivers\hidusb.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 18.50 KB (18,944 bytes), 3/25/2005 8:00 AM)

[Pointing Device]

Item	Value
Hardware Type	HID-compliant mouse
Number of Buttons	5
Status	OK
PNP Device ID	HID\VID_0624&PID_0296&MI_01&COL017&D5CFCD9&0&0000
Power Management Supported	No
Double Click Threshold	6
Handedness	Right Handed Operation
Driver	c:\windows\system32\drivers\mouhid.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 19.00 KB (19,456 bytes), 3/24/2005 1:21 PM)

Hardware Type	USB Human Interface Device
Number of Buttons	5
Status	OK
PNP Device ID	USB\VID_04B3&PID_4001&MI_01&6&2E1ACEF8&0&0001
Power Management Supported	No
Double Click Threshold	6
Handedness	Right Handed Operation
Driver	c:\windows\system32\drivers\hidusb.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 18.50 KB (18,944 bytes), 3/25/2005 8:00 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 8/28/2008 2:40 PM

Index 1

Service Name AsyncMac

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)

Adapter Type Not Available

Product Type WAN Miniport (L2TP)

Installed Yes

PNP Device ID ROOT\MS_L2TPMINIPOINT\0000

Last Reset 8/28/2008 2:40 PM

Index 2

Service Name Rasl2tp

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address Not Available

Driver c:\windows\system32\drivers\rasl2tp.sys

(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 132.00 KB

(135,168 bytes), 3/25/2005 8:00 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 8/28/2008 2:40 PM

Index 3

Service Name PptpMiniport

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address 50:50:54:50:30:30

Driver c:\windows\system32\drivers\rasppptp.sys

(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 117.50 KB

(120,320 bytes), 3/25/2005 8:00 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 8/28/2008 2:40 PM

Index 4

Service Name RasPppoe

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address 33:50:6F:45:30:30

Driver c:\windows\system32\drivers\rasppoe.sys

(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 67.50 KB

(69,120 bytes), 3/25/2005 8:00 AM)

Name [00000005] Direct Parallel

Adapter Type Not Available

Product Type Direct Parallel

Installed Yes

PNP Device ID ROOT\MS_PTMINIPOINT\0000

Last Reset 8/28/2008 2:40 PM

Index 5

Service Name Raspti

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address Not Available

Driver c:\windows\system32\drivers\raspti.sys

(5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 30.50 KB

(31,232 bytes), 3/25/2005 8:00 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 8/28/2008 2:40 PM

Index 6

Service Name NdisWan

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address Not Available

Driver c:\windows\system32\drivers\ndiswan.sys

(5.2.3790.3959 (srv03_sp2_rtm.070216-1710), 157.50 KB

(161,280 bytes), 3/25/2005 8:00 AM)

Name [00000007] Intel(R) PRO/1000 PT Dual Port Server

Adapter

Adapter Type Not Available

Product Type Intel(R) PRO/1000 PT Dual Port Server

Adapter

Installed Yes

PNP Device ID Not Available

Last Reset 8/28/2008 2:40 PM

Index 7

Service Name e1express

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No

DHCP Server Not Available

DHCP Lease Expires Not Available

DHCP Lease Obtained Not Available

MAC Address Not Available

Name [00000008] Intel(R) PRO/1000 PT Dual Port Server

Adapter

Adapter Type Not Available

Product Type Intel(R) PRO/1000 PT Dual Port Server

Adapter

Installed Yes

PNP Device ID Not Available

Last Reset 8/28/2008 2:40 PM

Index 8

Service Name e1express

IP Address Not Available

IP Subnet Not Available

Default IP Gateway Not Available

DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000009] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 9
Service Name e1express
IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000010] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 10
Service Name e1express
IP AddressNot 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 [00000011] Broadcom BCM5708C NetXtreme II
GigE (NDIS VBD Client)
Adapter Type Not Available
Product Type Broadcom BCM5708C NetXtreme II
GigE (NDIS VBD Client)
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 11
Service Name l2nd

IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000012] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 12
Service Name e1express
IP AddressNot 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] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 13
Service Name e1express
IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000014] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available

Last Reset 8/28/2008 2:40 PM
Index 14
Service Name e1express
IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000015] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 15
Service Name e1express
IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000016] Broadcom BCM5708C NetXtreme II
GigE (NDIS VBD Client)
Adapter Type Not Available
Product Type Broadcom BCM5708C NetXtreme II
GigE (NDIS VBD Client)
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:40 PM
Index 16
Service Name l2nd
IP AddressNot Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000017] Broadcom BCM5708C NetXtreme II
GigE (NDIS VBD Client)
Adapter Type Not Available
Product Type Broadcom BCM5708C NetXtreme II

GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 8/28/2008 2:40 PM
 Index 17
 Service Name l2nd
 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 [00000018] Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Adapter Type Not Available
 Product Type Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 8/28/2008 2:40 PM
 Index 18
 Service Name l2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000019] Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Adapter Type Not Available
 Product Type Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID Not Available
 Last Reset 8/28/2008 2:40 PM
 Index 19
 Service Name l2nd
 IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000020] Broadcom BCM5708C NetXtreme II

GigE (NDIS VBD Client)
 Adapter Type Ethernet 802.3
 Product Type Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID
 B06BDRVL2ND&PCI_164C14E4&SUBSYS_164C14E4&REV_12\6&33B9FAC3&0&20052B00
 Last Reset 8/28/2008 2:40 PM
 Index 20
 Service Name l2nd
 IP Address 10.0.0.1
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:10:18:25:48:EA
 Driver c:\windows\system32\drivers\bxnd52a.sys (3.7.6.0 built by: WinDDK, 64.00 KB (65,536 bytes), 6/30/2008 10:07 AM)

Name [00000021] Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Adapter Type Ethernet 802.3
 Product Type Broadcom BCM5708C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID
 B06BDRVL2ND&PCI_164C14E4&SUBSYS_164C14E4&REV_12\6&7B02679&0&20052500
 Last Reset 8/28/2008 2:40 PM
 Index 21
 Service Name l2nd
 IP Address 10.0.1.1
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:10:18:25:8C:64
 Driver c:\windows\system32\drivers\bxnd52a.sys (3.7.6.0 built by: WinDDK, 64.00 KB (65,536 bytes), 6/30/2008 10:07 AM)

Name [00000022] Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Adapter Type Ethernet 802.3
 Product Type Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID

B06BDRVL2ND&PCI_163914E4&SUBSYS_037C1014&REV_01\5&158780AB&0&20050200
 Last Reset 8/28/2008 2:40 PM
 Index 22
 Service Name l2nd
 IP Address 192.168.40.1
 IP Subnet 255.255.255.0
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:1A:64:35:20:38
 Driver c:\windows\system32\drivers\bxnd52a.sys (3.7.6.0 built by: WinDDK, 64.00 KB (65,536 bytes), 6/30/2008 10:07 AM)

Name [00000023] Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Adapter Type Ethernet 802.3
 Product Type Broadcom BCM5709C NetXtreme II
 GigE (NDIS VBD Client)
 Installed Yes
 PNP Device ID
 B06BDRVL2ND&PCI_163914E4&SUBSYS_037C1014&REV_01\5&38892308&0&20050200
 Last Reset 8/28/2008 2:40 PM
 Index 23
 Service Name l2nd
 IP Address 0.0.0.0
 IP Subnet 0.0.0.0
 Default IP Gateway Not Available
 DHCP Enabled Yes
 DHCP Server
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address 00:1A:64:35:20:3A
 Driver c:\windows\system32\drivers\bxnd52a.sys (3.7.6.0 built by: WinDDK, 64.00 KB (65,536 bytes), 6/30/2008 10:07 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 Tcip [UDP/IP]
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)

Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption No
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

Name RSVP UDP Service Provider
 Connectionless Service Yes
 Guarantees Delivery No
 Guarantees Sequencing No
 Maximum Address Size 16 bytes
 Maximum Message Size 63.93 KB (65,467 bytes)

Message Oriented Yes
 Minimum Address Size 16 bytes
 Pseudo Stream Oriented No
 Supports Broadcasting Yes
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data No
 Supports Graceful Closing No
 Supports Guaranteed Bandwidth No
 Supports Multicasting Yes

Name RSVP TCP Service Provider
 Connectionless Service No
 Guarantees Delivery Yes
 Guarantees Sequencing Yes
 Maximum Address Size 16 bytes
 Maximum Message Size 0 bytes
 Message Oriented No
 Minimum Address Size 16 bytes

Pseudo Stream Oriented No
 Supports Broadcasting No
 Supports Connect Data No
 Supports Disconnect Data No
 Supports Encryption Yes
 Supports Expedited Data Yes
 Supports Graceful Closing Yes
 Supports Guaranteed Bandwidth No
 Supports Multicasting No

[WinSock]

Item Value
 File c:\windows\system32\wsock32.dll
 Size 24.50 KB (25,088 bytes)
 Version 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)

[Ports]

[Serial]

Item Value
 Name Communications Port (COM1)
 Status OK
 PNP Device ID ACPI\PNP0501\1
 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 0x000003F8-0x000003FF
 IRQ Channel IRQ 4
 Driver c:\windows\system32\drivers\serial.sys
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 118.50 KB
 (121,344 bytes), 3/25/2005 8:00 AM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
 Drive C:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 68.36 GB (73,402,363,904 bytes)
 Free Space 47.88 GB (51,406,700,544 bytes)
 Volume Name
 Volume Serial Number 74B6D0C2
 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 No
 File System NTFS
 Size 31.00 MB (32,505,344 bytes)

Free Space 17.43 MB (18,280,448 bytes)
 Volume Name b21
 Volume Serial Number 1C5CA621

Drive G:
 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 X:
 Description Network Connection
 Provider Name \\fsserv\ray

Drive Y:
 Description Network Connection
 Provider Name \\fsserv\edrive

[Disks]

Item	Value
Description	Disk drive
Manufacturer	(Standard disk drives)
Model	IBM 1815 FAST SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	3
SCSI Bus	0
SCSI Logical Unit	0
SCSI Port	9
SCSI Target ID	0
Sectors/Track	63
Size	6.36 TB (6,995,493,711,360 bytes)
Total Cylinders	850,487
Total Sectors	13,663,073,655
Total Tracks	216,874,185
Tracks/Cylinder	255
Partition	Disk #22, Partition #0
Partition Size	225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #22, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #22, Partition #2
 Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	IBM 1815 FAST SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	3
SCSI Bus	0
SCSI Logical Unit	1
SCSI Port	9
SCSI Target ID	1
Sectors/Track	63
Size	6.36 TB (6,995,493,711,360 bytes)
Total Cylinders	850,487
Total Sectors	13,663,073,655
Total Tracks	216,874,185
Tracks/Cylinder	255
Partition	Disk #23, Partition #0
Partition Size	225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #23, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #23, Partition #2
 Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	IBM 1815 FAST SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	3
SCSI Bus	0
SCSI Logical Unit	0
SCSI Port	4
SCSI Target ID	0
Sectors/Track	63
Size	3.13 TB (3,442,880,125,440 bytes)
Total Cylinders	418,573
Total Sectors	6,724,375,245
Total Tracks	106,736,115
Tracks/Cylinder	255
Partition	Disk #4, Partition #0
Partition Size	225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #4, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #4, Partition #2
 Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	IBM 1815 FAST SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	3
SCSI Bus	0
SCSI Logical Unit	1
SCSI Port	4
SCSI Target ID	1
Sectors/Track	63
Size	3.13 TB (3,442,880,125,440 bytes)
Total Cylinders	418,573
Total Sectors	6,724,375,245
Total Tracks	106,736,115
Tracks/Cylinder	255
Partition	Disk #5, Partition #0
Partition Size	225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #5, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #5, Partition #2
 Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description	Disk drive
Manufacturer	(Standard disk drives)
Model	IBM 1815 FAST SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	2
SCSI Bus	0
SCSI Logical Unit	0
SCSI Port	4

SCSI Target ID 2
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #6, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #6, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 2
 SCSI Port 4
 SCSI Target ID 2
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #7, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 4
 SCSI Port 4

SCSI Target ID 2
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #8, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 6
 SCSI Port 4
 SCSI Target ID 2
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #9, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #9, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 8
 SCSI Port 4

SCSI Target ID 2
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #10, Partition #0
 Partition Size 1.13 TB (1,243,265,105,920 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #10, Partition #1
 Partition Size 1.13 TB (1,243,265,105,920 bytes)

Partition Starting Offset 1,243,399,341,056 bytes

Partition Disk #10, Partition #2
 Partition Size 31.00 MB (32,505,856 bytes)
 Partition Starting Offset 2,486,664,446,976 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 8
 SCSI Target ID 0
 Sectors/Track 63
 Size 6.36 TB (6,995,493,711,360 bytes)
 Total Cylinders 850,487
 Total Sectors 13,663,073,655
 Total Tracks 216,874,185
 Tracks/Cylinder 255
 Partition Disk #20, Partition #0
 Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #20, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #20, Partition #2
 Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive

Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 8
 SCSI Target ID 1
 Sectors/Track 63
 Size 6.36 TB (6,995,493,711,360 bytes)
 Total Cylinders 850,487
 Total Sectors 13,663,073,655
 Total Tracks 216,874,185
 Tracks/Cylinder 255
 Partition Disk #21, Partition #0
 Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #21, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #21, Partition #2
 Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #11, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 3
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #12, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #12, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 5
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #13, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #13, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 7
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #14, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #14, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASTT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 0
 SCSI Logical Unit 9
 SCSI Port 5
 SCSI Target ID 0
 Sectors/Track 63
 Size 2.26 TB (2,486,691,325,440 bytes)
 Total Cylinders 302,323
 Total Sectors 4,856,818,995
 Total Tracks 77,092,365
 Tracks/Cylinder 255
 Partition Disk #15, Partition #0
 Partition Size 1.13 TB (1,243,280,834,560 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #15, Partition #1
 Partition Size 1.13 TB (1,243,281,883,136 bytes)

Partition Starting Offset 1,243,415,069,696 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model IBM 1815 FAST SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63
Size 6.36 TB (6,995,493,711,360 bytes)
Total Cylinders 850,487
Total Sectors 13,663,073,655
Total Tracks 216,874,185
Tracks/Cylinder 255
Partition Disk #18, Partition #0
Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
Partition Disk #18, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #18, Partition #2
Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model IBM 1815 FAST SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63
Size 6.36 TB (6,995,493,711,360 bytes)
Total Cylinders 850,487
Total Sectors 13,663,073,655
Total Tracks 216,874,185
Tracks/Cylinder 255
Partition Disk #19, Partition #0

Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
Partition Disk #19, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #19, Partition #2
Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model IBM 1815 FAST SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 2
SCSI Target ID 0
Sectors/Track 63
Size 3.13 TB (3,442,880,125,440 bytes)
Total Cylinders 418,573
Total Sectors 6,724,375,245
Total Tracks 106,736,115
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
Partition Disk #0, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #0, Partition #2
Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Partition Starting Offset 134,235,136 bytes
Partition Disk #0, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #0, Partition #2
Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model IBM 1815 FAST SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0

SCSI Logical Unit 1
SCSI Port 2
SCSI Target ID 1
Sectors/Track 63
Size 3.13 TB (3,442,880,125,440 bytes)
Total Cylinders 418,573
Total Sectors 6,724,375,245
Total Tracks 106,736,115
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
Partition Disk #1, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #1, Partition #2
Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model IBM 1815 FAST SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 1
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 6.36 TB (6,995,493,711,360 bytes)
Total Cylinders 850,487
Total Sectors 13,663,073,655
Total Tracks 216,874,185
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
Partition Disk #16, Partition #1
Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #16, Partition #2
Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASiT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 6
 SCSI Target ID 1
 Sectors/Track 63
 Size 6.36 TB (6,995,493,711,360 bytes)
 Total Cylinders 850,487
 Total Sectors 13,663,073,655
 Total Tracks 216,874,185
 Tracks/Cylinder 255
 Partition Disk #17, Partition #0
 Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #17, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #17, Partition #2
 Partition Size 6.02 TB (6,622,066,769,920 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASiT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 3
 SCSI Target ID 0
 Sectors/Track 63
 Size 3.13 TB (3,442,880,125,440 bytes)
 Total Cylinders 418,573
 Total Sectors 6,724,375,245
 Total Tracks 106,736,115
 Tracks/Cylinder 255
 Partition Disk #2, Partition #0
 Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #2, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #2, Partition #2
 Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM 1815 FASiT SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 1
 SCSI Port 3
 SCSI Target ID 1
 Sectors/Track 63
 Size 3.13 TB (3,442,880,125,440 bytes)
 Total Cylinders 418,573
 Total Sectors 6,724,375,245
 Total Tracks 106,736,115
 Tracks/Cylinder 255
 Partition Disk #3, Partition #0
 Partition Size 225.10 GB (241,696,768,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #3, Partition #1
 Partition Size 122.56 GB (131,596,288,000 bytes)

Partition Starting Offset 241,831,003,136 bytes

Partition Disk #3, Partition #2
 Partition Size 2.79 TB (3,069,457,727,488 bytes)

Partition Starting Offset 373,427,291,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM ServeRAID-MR10M SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 2
 SCSI Bus 1
 SCSI Logical Unit 0

SCSI Port 11
 SCSI Target ID 0
 Sectors/Track 63
 Size 3.63 TB (3,991,991,592,960 bytes)
 Total Cylinders 485,332
 Total Sectors 7,796,858,580
 Total Tracks 123,759,660
 Tracks/Cylinder 255
 Partition Disk #25, Partition #0
 Partition Size 1.91 TB (2,097,676,288,000 bytes)

Partition Starting Offset 134,235,136 bytes
 Partition Disk #25, Partition #1
 Partition Size 488.28 GB (524,288,000,000 bytes)

Partition Starting Offset 2,097,810,523,136 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model IBM-ESXS ST973401SS 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 10
 SCSI Target ID 15
 Sectors/Track 63
 Size 68.36 GB (73,402,398,720 bytes)
 Total Cylinders 8,924
 Total Sectors 143,364,060
 Total Tracks 2,275,620
 Tracks/Cylinder 255
 Partition Disk #24, Partition #0
 Partition Size 68.36 GB (73,402,366,464 bytes)
 Partition Starting Offset 32,256 bytes

[SCSI]

Item Value
 Name LSI Adapter, SAS RAID-on-Chip, 8-port with 1078 - StorPort
 Manufacturer LSI Corporation
 Status OK
 PNP Device ID PCI\VEN_1000&DEV_0062&SUBSYS_10001000&REV_03\4&2BCCA3F5&0&0000
 Memory Address 0xEC550000-0xEC553FFF
 I/O Port 0x00003900-0x000039FF
 Memory Address 0xEC500000-0xEC5FFFFF
 IRQ Channel IRQ 46
 Driver c:\windows\system32\drivers\lsi_sas.sys (1.27.03.00)

built by: WinDDK, 122.50 KB (125,440 bytes), 5/24/2007 1:48 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&1DD35BBD&0&0000
I/O Port 0x00003A00-0x00003BFF
Memory Address 0xEC640000-0xEC643FFF
IRQ Channel IRQ 79
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&1DD35BBD&0&0100
I/O Port 0x00003B00-0x00003BFF
Memory Address 0xEC644000-0xEC647FFF
IRQ Channel IRQ 83
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&15BF98E9&0&0000
I/O Port 0x00003C00-0x00003DFF
Memory Address 0xEC740000-0xEC743FFF
IRQ Channel IRQ 80
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&15BF98E9&0&0100
I/O Port 0x00003D00-0x00003DFF
Memory Address 0xEC744000-0xEC747FFF
IRQ Channel IRQ 84
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name IBM ServeRAID-MR10M SAS/SATA Controller
Manufacturer LSI Corp.,
Status OK
PNP Device ID
PCI\VEN_1000&DEV_0060&SUBSYS_03791014&
REV_04\4&1DD220B0&0&0000
Memory Address 0xEC200000-0xEC2FFFFF
I/O Port 0x00003400-0x000034FF
Memory Address 0xEC240000-0xEC27FFFF
IRQ Channel IRQ 43
Driver c:\windows\system32\drivers\msas2k3.sys (2.21.0.64
built by: WinDDK, 26.50 KB (27,136 bytes), 5/25/2007 3:02 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&167C2C2F&0&0000
I/O Port 0x00003500-0x000036FF
Memory Address 0xEC340000-0xEC343FFF
IRQ Channel IRQ 44
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&167C2C2F&0&0100
I/O Port 0x00003600-0x000036FF
Memory Address 0xEC344000-0xEC347FFF
IRQ Channel IRQ 48
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&2657B4F5&0&0000
I/O Port 0x00003700-0x000038FF
Memory Address 0xEC440000-0xEC443FFF
IRQ Channel IRQ 45
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK
PNP Device ID
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&2657B4F5&0&0100
I/O Port 0x00003800-0x000038FF
Memory Address 0xEC444000-0xEC447FFF
IRQ Channel IRQ 49
Driver c:\windows\system32\drivers\ql2300.sys
(9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes),
5/25/2007 3:47 PM)

Name QLogic Optimizing and Multipath Driver
Manufacturer QLogic
Status Degraded
PNP Device ID ROOT\SCSIADAPTER\0000
Driver c:\windows\system32\drivers\qldirect.sys (8.01.13
Beta 3 (wx64), 62.00 KB (63,488 bytes), 5/29/2007 11:35 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_27DF&SUBSYS_03811014
&REV_01\3&29D18217&0&F9
I/O Port 0x00000700-0x0000070F
Driver c:\windows\system32\drivers\pciide.sys
(5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 6.00 KB (6,144
bytes), 3/25/2005 8:00 AM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)

Status OK
PNP Device ID
PCI\IDE\IDECHANNEL\4&37BDD80B&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), 146.50 KB
(150,016 bytes), 3/25/2005 8:00 AM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)

Status OK
 PNP Device ID
 PCI\IDE\DECHANNEL\4&37BDD80B&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), 146.50 KB
 (150,016 bytes), 3/25/2005 8:00 AM)

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code
--------	---------------	------------

[USB]

Device	PNP Device ID
Standard Universal PCI to USB Host Controller	PCI\VEN_8086&DEV_27C8&SUBSYS_03811014&REV_01\3&29D18217&0&E8
Standard Universal PCI to USB Host Controller	PCI\VEN_8086&DEV_27C9&SUBSYS_03811014&REV_01\3&29D18217&0&E9
Standard Universal PCI to USB Host Controller	PCI\VEN_8086&DEV_27CA&SUBSYS_03811014&REV_01\3&29D18217&0&EA
Standard Universal PCI to USB Host Controller	PCI\VEN_8086&DEV_27CB&SUBSYS_03811014&REV_01\3&29D18217&0&EB
Standard Enhanced PCI to USB Host Controller	PCI\VEN_8086&DEV_27CC&SUBSYS_03811014&REV_01\3&29D18217&0&EF

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started
		Start Mode	Status	Error Control
		Accept Pause	Accept Stop	
abiosdsk	Abiosdsk	Not Available	Kernel Driver	
		No	Disabled Stopped	OK Ignore
acpi	Microsoft ACPI Driver			
		c:\windows\system32\drivers\acpi.sys	Kernel	
Driver	Yes	Boot	Running	OK Normal
	No	Yes		
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys		

	Kernel Driver	No	Disabled	Stopped	
	OK	Normal	No	No	
adpu160m	adpu160m	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
adpu320	adpu320	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
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 Normal	
		No	No		
aic78xx	aic78xx	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
aliide	AliIde	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
amdide	AmdIde	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
arc	arc	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Normal	
		No	No		
asynmac	RAS Asynchronous Media Driver				
		c:\windows\system32\drivers\asynmac.sys	Kernel		
Driver	No	Manual	Stopped	OK Normal	
	No	No			
atapi	Standard IDE/ESDI Hard Disk Controller				
		c:\windows\system32\drivers\atapi.sys	Kernel		
Driver	Yes	Boot	Running	OK Normal	
	No	Yes			
atdisk	Atdisk	Not Available	Kernel Driver		
		No	Disabled Stopped	OK Ignore	
		No	No		
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	Broadcom NetXtreme II VBD				
		c:\windows\system32\drivers\bxvbda.sys	Kernel		
Driver	Yes	Boot	Running	OK Normal	
	No	Yes			
b06diag	Broadcom NetXtreme II Diag Driver				
		c:\windows\system32\drivers\bxdiaga.sys	Kernel		

Driver	No	Manual	Stopped	OK	Normal
	No	No			
beep	Beep	c:\windows\system32\drivers\beep.sys			
	Kernel Driver	Yes	System	Running	
	OK	Normal	No	Yes	
cdac15ba	CdaC15BA				
		c:\windows\system32\drivers\cdac15ba.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
cdad10ba	CdaD10BA				
		c:\windows\system32\drivers\cdad10ba.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
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	Ignore
	No	No			
clusdisk	Cluster Disk Driver				
		c:\windows\system32\drivers\clusdisk.sys	Kernel		
Driver	No	Disabled	Stopped	OK	Normal
	No	No			
cmdide	CmdIde	Not Available	Kernel Driver		
	No	Disabled	Stopped	OK	Normal
	No	No			
cpqccism	cpqccism	Not Available	Kernel Driver		
	No	Disabled	Stopped	OK	Normal
	No	No			
cpuz128	cpuz128				
		\\?\c:\docume~1\admini~1\locals~1\temp\cpuz_x64.s			
ys	Kernel Driver	No	Manual	Stopped	
	OK	Normal	No	No	
crcdisk	CRC Disk Filter Driver				
		c:\windows\system32\drivers\crcdisk.sys	Kernel		
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
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

dmload	No	Yes			
	Kernel Driver	Yes	Boot	Running	
dpti2o	OK	Normal	No	Yes	
	Kernel Driver	Not Available	Stopped	OK	Normal
dsapmem	No	Disabled	Stopped	OK	Normal
	Kernel Driver	Not Available	Stopped	OK	Normal
Driver	No	Manual	Stopped	OK	Ignore
elexpress	Intel(R) PRO/1000 PCI Express Network Connection				
Driver	c:\windows\system32\drivers\ele5132e.sys	Kernel Driver	No	Manual	Stopped
Driver	OK	Normal	Stopped	OK	Normal
elxstor	No	Not Available	Stopped	OK	Normal
	Kernel Driver	Not Available	Stopped	OK	Normal
em	No	Disabled	Stopped	OK	Normal
	Kernel Driver	Not Available	Stopped	OK	Normal
fastfat	em	\\?\c:\windows\system32\drivers\em.sys	Kernel Driver	No	Manual
	OK	Normal	No	No	Stopped
fdisk	Fastfat	c:\windows\system32\drivers\fastfat.sys	File System Driver	No	Disabled
	OK	Normal	No	No	Stopped
fips	Fdc	c:\windows\system32\drivers\fdc.sys	Kernel Driver	No	System
	OK	Ignore	No	No	Stopped
flpydisk	Fips	c:\windows\system32\drivers\lfips.sys	Kernel Driver	Yes	System
	OK	Normal	No	Yes	Running
fltmgr	Flpydisk	c:\windows\system32\drivers\flpydisk.sys	Kernel Driver	No	System
	OK	Ignore	No	No	Stopped
ftdisk	FltMgr	c:\windows\system32\drivers\fltmgr.sys	File System Driver	Yes	Boot
	OK	Normal	No	Yes	Running
gpc	ftdisk	Volume Manager Driver	c:\windows\system32\drivers\ftdisk.sys	Kernel Driver	Yes
	Driver	Yes	Boot	Running	OK
hidusb	gpc	Generic Packet Classifier	c:\windows\system32\drivers\msgpc.sys	Kernel Driver	No
	Driver	Yes	Manual	Running	OK
hpcisss	hidusb	Microsoft HID Class Driver	c:\windows\system32\drivers\hidusb.sys	Kernel Driver	Yes
	Driver	Yes	Manual	Running	OK
http	hpcisss	Not Available	Kernel Driver	No	Disabled
	Driver	No	Disabled	Stopped	OK
	Kernel Driver	No	Manual	Stopped	Ignore
	OK	Normal	No	No	Normal

i2omgmt	i2omgmt	Not Available	Kernel Driver	No	System
	OK	Stopped	OK	Normal	
i8042prt	i8042prt	c:\windows\system32\drivers\i8042prt.sys	Kernel Driver	No	System
	OK	Ignore	No	No	Stopped
iirsp	iirsp	Not Available	Kernel Driver	No	Disabled
	Driver	No	Disabled	Stopped	OK
imapi	CD-Burning Filter Driver		c:\windows\system32\drivers\imapi.sys	Kernel Driver	Yes
	Driver	Yes	System	Running	OK
intelide	IntelIde	Not Available	Kernel Driver	No	Yes
	Driver	No	Disabled	Stopped	OK
intelppm	Intel Processor Driver		c:\windows\system32\drivers\intelppm.sys	Kernel Driver	Yes
	Driver	Yes	Manual	Running	OK
iomonkey	IoMonkey		\\?\c:\docume~1\admini~1\locals~1\temp\iomonkey.s	Kernel Driver	No
ip6fw	IPV6 Windows Firewall Driver		c:\windows\system32\drivers\ip6fw.sys	Kernel Driver	No
	Driver	No	Manual	Stopped	OK
ipfilterdriver	IP Traffic Filter Driver		c:\windows\system32\drivers\ipfltdrv.sys	Kernel Driver	No
	Driver	No	Manual	Stopped	OK
ipinip	IP in IP Tunnel Driver		c:\windows\system32\drivers\ipinip.sys	Kernel Driver	No
	Driver	No	Manual	Stopped	OK
ipnat	IP Network Address Translator		c:\windows\system32\drivers\ipnat.sys	Kernel Driver	No
	Driver	No	Manual	Stopped	OK
ipsec	IPSEC driver		c:\windows\system32\drivers\ipsec.sys	Kernel Driver	No
	Driver	Yes	System	Running	OK
isapnp	PnP ISA/EISA Bus Driver		c:\windows\system32\drivers\isapnp.sys	Kernel Driver	Yes
	Driver	Yes	Boot	Running	OK
kbdclass	Keyboard Class Driver		c:\windows\system32\drivers\kbdclass.sys	Kernel Driver	No
	Driver	Yes	System	Running	OK
kbdhid	Keyboard HID Driver		c:\windows\system32\drivers\kbdhid.sys	Kernel Driver	No

Driver	Yes	System	Running	OK	Ignore
ksecdd	KSecDD	c:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot
	OK	Normal	No	Yes	Running
ksthunk	Kernel Streaming WOW64 Thunk Service		c:\windows\system32\drivers\ksthunk.sys	Kernel Driver	Yes
	Driver	Yes	Manual	Running	OK
l2nd	Broadcom NetXtreme II BXND		c:\windows\system32\drivers\bxnd52a.sys	Kernel Driver	No
	Driver	Yes	Manual	Running	OK
lp6nds35	Ip6nds35	Not Available	Kernel Driver	No	Yes
	Driver	No	Disabled	Stopped	OK
lsi_sas	lsi_sas	c:\windows\system32\drivers\lsi_sas.sys	Kernel Driver	Yes	Boot
	Driver	OK	Normal	No	Yes
mnmdd	mnmdd	c:\windows\system32\drivers\mnmdd.sys	Kernel Driver	Yes	System
	Driver	OK	Ignore	No	Yes
modem	Modem	c:\windows\system32\drivers\modem.sys	Kernel Driver	No	Manual
	Driver	OK	Ignore	No	Stopped
mouclass	Mouse Class Driver		c:\windows\system32\drivers\mouclass.sys	Kernel Driver	Yes
	Driver	Yes	System	Running	OK
mouhid	Mouse HID Driver		c:\windows\system32\drivers\mouhid.sys	Kernel Driver	Yes
	Driver	Yes	Manual	Running	OK
mountmgr	Mount Point Manager		c:\windows\system32\drivers\mountmgr.sys	Kernel Driver	Yes
	Driver	Yes	Boot	Running	OK
mrraid35x	mrraid35x	Not Available	Kernel Driver	No	Yes
	Driver	No	Disabled	Stopped	OK
mrxdav	WebDav Client Redirector		c:\windows\system32\drivers\mrxdav.sys	File System Driver	No
	System Driver	Normal	No	Manual	Stopped
mrxsmmb	MRXSMB	c:\windows\system32\drivers\mrxsmmb.sys	File System Driver	Yes	System
	Driver	OK	Normal	No	Yes
msas2k3	msas2k3	c:\windows\system32\drivers\msas2k3.sys	Kernel Driver	Yes	Boot
	Driver	OK	Normal	No	Yes
msfs	Msfs	c:\windows\system32\drivers\msfs.sys	File System Driver	Yes	System
	Driver	OK	Normal	No	Yes
mssmbios	Microsoft System Management BIOS Driver				

Driver	c:\windows\system32\drivers\mssmbios.sys	Kernel			
	Yes	Manual	Running	OK	Normal
	No	Yes			
mup	Mup	c:\windows\system32\drivers\mup.sys			
	File System Driver	Yes	Boot	Running	
	OK	Normal	No	Yes	
ndis	NDIS System Driver				
	c:\windows\system32\drivers\ndis.sys	Kernel			
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
ndistapi	Remote Access NDIS TAPI Driver				
	c:\windows\system32\drivers\ndistapi.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ndisuio	NDIS Usermode I/O Protocol				
	c:\windows\system32\drivers\ndisuio.sys	Kernel			
Driver	No	Manual	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			
ndproxy	NDIS Proxy				
	c:\windows\system32\drivers\ndproxy.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
netbios	NetBIOS Interface				
	c:\windows\system32\drivers\netbios.sys	File			
System Driver	Yes	System	Running	OK	
	Normal	No	Yes		
netbt	NetBios over Tcpip				
	c:\windows\system32\drivers\netbt.sys	Kernel			
Driver	Yes	System	Running	OK	Normal
	No	Yes			
nfrd960	nfrd960	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
npfs	Npfs	c:\windows\system32\drivers\npfs.sys			
	File System Driver	Yes	System	Running	
	OK	Normal	No	Yes	
ntfs	Ntfs	c:\windows\system32\drivers\ntfs.sys			
	File System Driver	Yes	Disabled	Running	
	OK	Normal	No	Yes	
null	Null	c:\windows\system32\drivers\null.sys			
	Kernel Driver	Yes	System	Running	
	OK	Normal	No	Yes	
parport	Parport	c:\windows\system32\drivers\parport.sys			
	Kernel Driver	No	Manual	Stopped	
	OK	Ignore	No	No	
partmgr	Partition Manager				
	c:\windows\system32\drivers\partmgr.sys	Kernel			
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
pci	PCI Bus Driver				

Driver	c:\windows\system32\drivers\pci.sys	Kernel			
	Yes	Boot	Running	OK	Critical
	No	Yes			
pciide	PCIIde	c:\windows\system32\drivers\pciide.sys			
	Kernel Driver	Yes	Boot	Running	
	OK	Normal	No	Yes	
pcmcia	Pcmcia	c:\windows\system32\drivers\pcmcia.sys			
	Kernel Driver	No	Disabled	Stopped	
	OK	Normal	No	No	
pdcomp	PDCOMP	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	Ignore
	No	No			
pdframe	PDFRAME	Not Available		Kernel	
Driver	No	Manual	Stopped	OK	Ignore
	No	No			
pdreli	PDRELI	Not Available		Kernel Driver	
	No	Manual	Stopped	OK	Ignore
	No	No			
pdrframe	PDRFRAME	Not Available		Kernel	
Driver	No	Manual	Stopped	OK	Ignore
	No	No			
pnpmem	Microsoft Memory Module Driver				
	c:\windows\system32\drivers\pnpmem.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
pptpminiport	WAN Miniport (PPTP)				
	c:\windows\system32\drivers\raspppt.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ptilink	Direct Parallel Link Driver				
	c:\windows\system32\drivers\ptilink.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
ql2300	QLogic Fibre Channel SCSI Miniport Driver (wx64 IP)				
	c:\windows\system32\drivers\ql2300.sys	Kernel			
Driver	Yes	Boot	Running	OK	Normal
	No	Yes			
qldirect	qldirect	c:\windows\system32\drivers\qldirect.sys			
	Kernel Driver	Yes	Auto	Running	
	OK	Normal	No	Yes	
rasacd	Remote Access Auto Connection Driver				
	c:\windows\system32\drivers\rasacd.sys	Kernel			
Driver	Yes	System	Running	OK	Normal
	No	Yes			
rasl2tp	WAN Miniport (L2TP)				
	c:\windows\system32\drivers\rasl2tp.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
raspppoe	Remote Access PPPOE Driver				
	c:\windows\system32\drivers\raspppoe.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
raspti	Direct Parallel				
	c:\windows\system32\drivers\raspti.sys	Kernel			

Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
rdcss	Rdcss	c:\windows\system32\drivers\rdcss.sys			
	File System Driver	Yes	System	Running	
	OK	Normal	No	Yes	
rdpcdd	RDPCDD	c:\windows\system32\drivers\rdpcdd.sys			
	Kernel Driver	Yes	System	Running	
	OK	Ignore	No	Yes	
rdpdr	Terminal Server Device Redirector Driver				
	c:\windows\system32\drivers\rdpdr.sys	Kernel			
Driver	Yes	Manual	Running	OK	Normal
	No	Yes			
rdpwd	RDPWD	c:\windows\system32\drivers\rdpwd.sys			
	Kernel Driver	Yes	Manual	Running	
	OK	Ignore	No	Yes	
redbook	Digital CD Audio Playback Filter Driver				
	c:\windows\system32\drivers\redbook.sys	Kernel			
Driver	Yes	System	Running	OK	Normal
	No	Yes			
rtcore64	RTCore64				
	\\?c:\[install]\rightmark_clockutil\rmclock_230_bin				
upd1\rtcore64.sys	Kernel Driver	No	Manual	Stopped	No
	Stopped	OK	Normal	No	No
secdrv	Security Driver				
	c:\windows\system32\drivers\secdrv.sys	Kernel			
Driver	No	Disabled	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			
setcpuservice	SetCPUService	\\?c:\program files			
(x86)\throttletoolx64\setcpuservice.sys	Kernel Driver				
	No	Manual	Stopped	OK	Ignore
	No	No			
sfloppy	High-Capacity Floppy Disk Drive				
	c:\windows\system32\drivers\sfloppy.sys	Kernel			
Driver	No	Manual	Stopped	OK	Normal
	No	No			
simbad	Simbad	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
sivdriver	SIV Kernel Driver				
	\\?c:\windows\system32\drivers\sivx64.sys	Kernel			
Driver	No	Manual	Stopped	OK	Normal
	No	No			
srv	Srv	c:\windows\system32\drivers\srv.sys			
	File System Driver	Yes	Manual	Running	
	OK	Normal	No	Yes	

swenum	Software Bus Driver				
Driver	Yes	Manual	Running	OK	Kernel Normal
	No	Yes			
symc8xx	symc8xx	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
symmpi	symmpi	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
sym_hi	sym_hi	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
sym_u3	sym_u3	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
tcpip	TCP/IP Protocol Driver				
Driver	Yes	System	Running	OK	Kernel Normal
	No	Yes			
tdpipe	TDPIPE	No	Manual	Stopped	Kernel Driver
	OK	Ignore	No	No	Normal
tdtcp	TDTCP	Yes	Manual	Running	Kernel Driver
	OK	Ignore	No	Yes	Normal
termdd	Terminal Device Driver				
Driver	Yes	System	Running	OK	Kernel Normal
	No	Yes			
toside	TosIde	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
udfs	Udfs	Disabled	Stopped		Kernel Driver
	OK	Normal	No	No	Normal
ultra	ultra	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
update	Microcode Update Driver				
Driver	Yes	Manual	Running	OK	Kernel Normal
	No	Yes			
usbccgp	Microsoft USB Generic Parent Driver				
Driver	Yes	Manual	Running	OK	Kernel Normal
	No	Yes			
usbhci	Microsoft USB 2.0 Enhanced Host Controller				
Miniport Driver	Yes	Manual	Running	OK	Kernel Driver
	OK	Normal	No	Yes	Normal
usbhub	USB2 Enabled Hub				
Driver	Yes	Manual	Running	OK	Kernel Normal

usbstor	No	Yes			
Driver	Yes	Manual	Stopped	OK	Kernel Normal
	No	No			
usbuhci	Microsoft USB Universal Host Controller				
Driver	Yes	Manual	Running	OK	Kernel Normal
	No	Yes			
vga	vga	No	Manual	Stopped	Kernel Driver
	OK	Ignore	No	No	Normal
vgasave	VGA Display Controller.				
Driver	Yes	System	Running	OK	Kernel Ignore
	No	Yes			
viaide	Vialde	Not Available			Kernel Driver
	No	Disabled	Stopped	OK	Normal
	No	No			
volsnap	Storage volumes				
Driver	Yes	Boot	Running	OK	Kernel Normal
	No	Yes			
wanarp	Remote Access IP ARP Driver				
Driver	Yes	Manual	Running	OK	Kernel Normal
	No	Yes			
wdf01000	Wdf01000				
Driver	Yes	Boot	Running	OK	Kernel Normal
	No	Yes			
wdica	WDICA	Not Available			Kernel Driver
	No	Manual	Stopped	OK	Ignore
	No	No			
wlbs	Network Load Balancing				
Driver	No	Manual	Stopped	OK	Kernel Normal
	No	No			
[Signed Drivers]					
Device Name	Signed	Device Class	Driver		
Version	Driver Date	Manufacturer	INF		
Name	Driver Name	Device ID			
Microsoft	System Management BIOS Driver	No			
	SYSTEM 5.2.3790.1830	10/1/2002			
	(Standard system devices)	machine.inf			
	Not Available	ROOT\SYSTEM\0002			
Microcode Update Device	No	SYSTEM			
5.2.3790.1830	10/1/2002	(Standard system			
devices)	machine.inf	Not Available			
	ROOT\SYSTEM\0001				
Plug and Play Software Device Enumerator	No				

SYSTEM	5.2.3790.1830	10/1/2002			
(Standard system devices)	machine.inf				
Not Available	ROOT\SYSTEM\0000				
QLogic Optimizing and Multipath Driver	No				
SCSIADAPTER	Not Available	Not			
Available	Not Available	oem9.inf			
	ROOT\SCSIADAPTER\0000				
Terminal Server Mouse Driver	No	SYSTEM			
5.2.3790.1830	10/1/2002	(Standard system			
devices)	machine.inf	Not Available			
	ROOT\RDP_MOU\0000				
Terminal Server Keyboard Driver	No				
SYSTEM 5.2.3790.1830	10/1/2002				
(Standard system devices)	machine.inf				
Not Available	ROOT\RDP_KBD\0000				
Terminal Server Device Redirector	No				
SYSTEM 5.2.3790.1830	10/1/2002				
(Standard system devices)	machine.inf				
	ROOT\RDPDR\0000				
Direct Parallel	No	NET			
10/1/2002	Microsoft	netrasa.inf			
	Not Available	ROOT\MS_PTMINIPORT\0000			
WAN Miniport (PPTP)	No	NET			
5.2.3790.1830	10/1/2002	Microsoft			
	netrasa.inf	Not Available			
	ROOT\MS_PPTMINIPORT\0000				
WAN Miniport (PPPOE)	No	NET			
5.2.3790.1830	10/1/2002	Microsoft			
	netrasa.inf	Not Available			
	ROOT\MS_PPPOEMINIPORT\0000				
WAN Miniport (IP)	No	NET			
10/1/2002	Microsoft	netrasa.inf			
	Not Available	ROOT\MS_NDISWANIP\0000			
WAN Miniport (L2TP)	No	NET			
5.2.3790.1830	10/1/2002	Microsoft			
	netrasa.inf	Not Available			
	ROOT\MS_L2TPMINIPORT\0000				
Video Codecs	No	MEDIA			
10/1/2002	(Standard system devices)	wave.inf			
Not Available	ROOT\MEDIA\MS_MMVID				
Legacy Video Capture Devices	No	MEDIA			
5.2.3790.1830	10/1/2002	(Standard system			
devices)	wave.inf	Not Available			
	ROOT\MEDIA\MS_MMVCD				
Media Control Devices	No	MEDIA			
5.2.3790.1830	10/1/2002	(Standard system			
devices)	wave.inf	Not Available			
	ROOT\MEDIA\MS_MMMCI				
Legacy Audio Drivers	No	MEDIA			
10/1/2002	(Standard system devices)	wave.inf			
Not Available	ROOT\MEDIA\MS_MMDRV				

Audio Codecs	No	MEDIA 5.2.3790.1830	Not Available	Not Available	Not Available	Not Available	Not Available	em	Not Available	LEGACYDRIVER	Not Available	Not Available
	10/1/2002 (Standard system devices)	wave.inf	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
	Not Available	ROOT\MEDIA\MS_MM\ACM	Null	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	ROOT\LEGACY_EM\0000	Not Available	Not Available
Wdf01000	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	DSAPMem	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	NetBios over Tcpip	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_WDF01000\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_DSAPMEM\0000	Not Available	Not Available
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	dmload	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available			ROOT\LEGACY_NETBT\0000	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	NDProxy	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_DMLOAD\0000	Not Available	Not Available
volsnap	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	dmboot	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available			ROOT\LEGACY_NDPROXY\0000	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_VOLS\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_DMBOOT\0000	Not Available	Not Available
VGA Display Controller	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	CRC Disk	Filter Driver	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	ROOT\LEGACY_CRCDISK\0000	Not Available	Not Available	Not Available
TDTCP	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	cpuz128	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	NDIS System Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_TDTCP\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_CPUZ128\0000	Not Available	Not Available
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	CdaD10BA	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	mountmgr	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
SIV Kernel Driver	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_CDAD10BA\0000	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	CdaC15BA	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_SIVDRIVER\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_CDAC15BA\0000	Not Available	Not Available
Security Driver	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	Beep	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	ksecdd	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_SECDRV\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_BEEP\0000	Not Available	Not Available
RTCORE64	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	AFD	Not Available	LEGACYDRIVER	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	IPSEC driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available
		ROOT\LEGACY_RTCORE64\0000	Available	Not Available	Not Available	Not Available	Not Available			ROOT\LEGACY_AFD\0000	Not Available	Not Available
RDPWD	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	Generic volume	No	VOLUME 5.2.3790.1830	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available			10/1/2002 Microsoft volume.inf	Not Available	Not Available
Available	Not Available	Not Available	IP Network Address Translator	Not Available	LEGACYDRIVER	Not Available	Not Available			STORAGE\VOLUME\1&30A96598&0&GPTPART	Not Available	Not Available
		ROOT\LEGACY_RDPWD\0000	Available	Not Available	Not Available	Not Available	Not Available	ITION	{306901D5-8EEB-4152-9F63-39575283705B}	Not Available	Not Available	Not Available
RDPD	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	Generic volume	No	VOLUME 5.2.3790.1830	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available			10/1/2002 Microsoft volume.inf	Not Available	Not Available
Available	Not Available	Not Available	Generic Packet Classifier	Not Available	LEGACYDRIVER	Not Available	Not Available			STORAGE\VOLUME\1&30A96598&0&GPTPART	Not Available	Not Available
		ROOT\LEGACY_RDPD\0000	Available	Not Available	Not Available	Not Available	Not Available	ITION	{5B022651-8D8F-41A5-8A6F-08FC569518B5}	Not Available	Not Available	Not Available
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	Generic volume	No	VOLUME 5.2.3790.1830	Not Available	Not Available
Available	Not Available	Not Available	Available	Not Available	Not Available	Not Available	Not Available			10/1/2002 Microsoft volume.inf	Not Available	Not Available
Available	Not Available	Not Available	Fips	Not Available	LEGACYDRIVER	Not Available	Not Available			STORAGE\VOLUME\1&30A96598&0&GPTPART	Not Available	Not Available
		ROOT\LEGACY_RASACD\0000	Available	Not Available	Not Available	Not Available	Not Available	ITION	{17F5461B-66FA-48F5-80D9-913ED251DF2F}	Not Available	Not Available	Not Available
Partition Manager	Not Available	LEGACYDRIVER	Available	Not Available	Not Available	Not Available	Not Available	Generic volume	No	VOLUME 5.2.3790.1830	Not Available	Not Available
		ROOT\LEGACY_PARTITION\0000	Available	Not Available	Not Available	Not Available	Not Available			10/1/2002 Microsoft volume.inf	Not Available	Not Available

10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{8951A455-B7A1-4566-BC7D-AAEC17BA5180}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{FB6181E9-CC1C-4A75-89C6-49F9C6C0852A}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{8DCFE286-5F0A-443E-9EA3-91ACED79CD37}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{019F8817-73EB-45A3-B97A-791F56A8C04A}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{5A3F5E56-74C6-4E1F-B5BA-35618D21A20C}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{43C6BF6F-FFDF-4121-92BD-280309A3DBD8}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{1F106E4D-9334-4126-AC5E-2C3B97D5F626}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{F2DFF9A1-DCD8-464A-9AFA-1E0A46323308}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{33071E86-6BBF-429B-BD2D-19824BF773F3}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{3EC9D580-B936-49C1-8E42-879241B39B5C}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{0D683E57-7686-4A60-8C9D-49197D545BD3}

Generic volume No VOLUME 5.2.3790.1830

10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{6213D85D-CC95-4EED-84E0-D1D508DC3DFC}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{21AD24AE-DFD2-4DD1-ABC4-64F07B1EF6A2}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{558D683D-6EE4-48B6-AFE3-9E8E4314DC10}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{3E5DDF4B-787C-46D5-9484-E8A055B823D6}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{B27C6B87-9059-425D-B24B-166C43896E51}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{97DA0EA0-1894-4E31-9102-FF5F64D953FB}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{D76F1F89-2259-47AC-AA71-6502E8123BD7}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{98939175-3F1A-4781-A647-68B5E8B94513}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{4529BA6C-5924-4F74-99CC-7118D7E6FC0D}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{BB066F59-E9FB-42F6-9E5B-7E1983ABCD3D}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{EBE84B1C-8A5D-49A3-B5C6-65C77F16F3B8}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available

STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{C0AEE114-F7F5-4031-86EE-ABF70DD8C75E}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{63E66BC7-BD74-4DC3-97FC-E60E374A1942}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{2357E495-989B-4AD6-B23F-DF198E162801}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{5A02BE7F-3AEA-4336-ABAC-DF07BF51FBA6}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{03F844A5-65AC-485A-8BB9-89CF8FC9B801}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{1E7890A1-0C2E-4F2E-92F3-9C78A174E978}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{DEA05250-19C2-4AA5-8205-99F7D25FDE7A}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{2CB25CAE-0F63-4693-864E-B42B2797965A}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{D085B96C-D526-43FB-A11F-93788C9BCE83}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{8EEA5E2B-0AF9-4732-B870-840A50E10EBE}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{35E10EA5-C6D8-456C-B2BF-2B074BA4BDFB}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART

ITION{7D4BF553-7A7F-4D26-9B51-BF27F6D1B891}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{05387FDA-ED74-4061-8BA0-8EDE63C63038}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{371E92CB-7D15-4FE0-A8F4-1B706289F748}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{11A53C54-268F-4388-A5C5-9404E400666C}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{B1353B0C-0F7F-4DA6-BF84-91265DBFFFAE}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{2EE4DE6D-C849-4C4D-83F5-F472D2FF7E55}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{EE7735AC-A55F-4FAF-9C17-4E4AE7D37F87}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{FBCC80F3-F52E-49E7-A062-A225DACF9C15}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{F59351A9-1744-4C06-AFCE-96FE82BE246E}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{F39E3776-2F45-41D1-8CCC-ED2FBF265E70}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{FA59D065-B040-42DA-8083-C2A3C7442264}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{7613CE62-470D-440E-A411-541D44048F8C}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{D3816E5F-8A17-4302-BFC8-83501E46262D}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{CAA8FCD9-6243-4B84-A810-B0DCAB1AF872}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{B8FAD3A4-797A-4FF9-85EA-5E3E919217E7}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{3E817A81-C5E8-465D-B4A2-C572C5C517C2}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{A858325D-EFA8-4177-A9B4-7A864561CC69}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{BB039E89-D711-4028-B5A3-AF4F59A6894E}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{2217FD15-06A9-4CE8-85A5-B7024A7BBEF3}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{6C43B08A-E14E-4130-BF8F-6F4BD6E7490E}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{9C4EBF25-12BE-4804-8BE1-C19E4E9D9184}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{C9921BC5-E90A-4F87-AB50-63AAB41A579C}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{9C04E894-4DF6-47D5-9BEC-F9127D2CC6ED}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{EB8E62D1-AF4A-471D-9B71-CA730E9ABE74}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{F32B9C75-3D7E-4B55-9B30-6201CB8C6A43}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{EF434EE9-2CD4-450A-BA75-06A148787CE8}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{A87DF905-DF4A-41FC-B3EF-C6FDD08E684B}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{431103CD-1479-4781-9310-8BA2815C4D20}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{788C7768-FA96-4939-AC19-67FD0720614A}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{14D562C5-4B8A-4FF1-A633-5E8E8C578ED2}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{5F0E5F8C-2B7D-4884-98BF-4D76588716CF}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{8E0313D0-0406-4A0C-A413-F9C33117E590}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{FA5CAED7-8D2D-4747-8DE9-2E08ACF575D0}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART
ITION{4A520BFF-5244-46E4-8C2A-939C2D173E45}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf/Not Available
STORAGE\VOLUME\1&30A96598&0&GPTPART

ITION{0EEAAB2C-92A7-4A2E-A599-551BEF131CBA}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{4E3CF05E-7860-41C4-8F5E-2468B4CDAFB9}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{D1938028-0829-4C9D-93AA-DFD5E60A14FB}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{53040945-5566-44E5-A7D5-D46972C982D9}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{33CCB3BB-F07A-4917-8E41-1784C2A04AF2}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{0B87EA56-1248-4D4C-AA51-8DBECC9358C7}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{CB5E897A-767C-45E2-9722-FB0C61D85568}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{AE3B25C1-4992-4CB4-BB04-1213160E9850}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{6BEF39E5-CA1A-4206-84AD-BD4B816FEC54}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{9576C25C-9025-441F-9DAA-5F83D0C3AB58}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{5458BE0C-E84A-46C0-BD12-208B0F39A133}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{2137FC0A-DF31-4D69-9C76-9E0B48621925}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{2DA9DAF6-7969-41D7-A56A-BE044A703300}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{8613DE85-9FF7-4463-8585-9F695527BF0E}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{E7C588B6-B5E8-4629-974E-0CCAE4D9017D}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{43942844-29A5-4D1C-8D1E-9B6FD4457CDD}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{CD0FF311-43CC-4D0E-96B7-4BEA3C755670}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{9E386F9F-1B6A-4FB1-ABE3-B1F163503623}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{5226B600-B501-4CB7-800F-E6E7B3D1215D}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&GPTPART
ITION{75AD8103-809A-4328-8DBA-27D62DA89369}

Generic volume No VOLUME 5.2.3790.1830
10/1/2002 Microsoft volume.inf Not Available
STORAGE\VOLUME1\&30A96598&0&SIGNATU
RE97F497F4OFFSET7E00LENGTH11171F3A00

Volume Manager No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ROOT\FTDISK\0000

Logical Disk Manager No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ROOT\DMIO\0000

Qlogic processor device No SYSTEM
5.2.3790.1830 10/1/2002 QLOGIC
scsidev.inf Not Available

SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
DO_DEVICE&REV_15&A890E61&0&007F00

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&A890E61&0&011

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&A890E61&0&000

QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.0.12 9/21/2005 QLogic oem7.inf Not
Available

PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_024&2657B4F5&0&0100

Qlogic processor device No SYSTEM
5.2.3790.1830 10/1/2002 QLOGIC
scsidev.inf Not Available

SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
DO_DEVICE&REV_15&2D8AB0BE&0&007F00

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&2D8AB0BE&0&009

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&2D8AB0BE&0&007

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&2D8AB0BE&0&005

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&2D8AB0BE&0&003

Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_09145&2D8AB0BE&0&001

QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.0.12 9/21/2005 QLogic oem7.inf Not
Available

PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_024&2657B4F5&0&0000

PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3790.1830 10/1/2002 (Standard system

devices) machine.inf Not Available
 PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
 REV_013&2FF92BID&0&00
 PCI bus No SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0A08\7
 Qlogic processor device No SYSTEM
 5.2.3790.1830 10/1/2002 QLOGIC
 scsidev.inf Not Available
 SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
 DO_DEVICE&REV_5&2F170704&0&007F00
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&2F170704&0&011
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&2F170704&0&000
 QLogic Fibre Channel Adapter No SCSIADAPTER
 9.1.0.12 9/21/2005 QLogic oem7.inf Not
 Available
 PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
 REV_02\4&167C2C2F&0&0100
 Qlogic processor device No SYSTEM
 5.2.3790.1830 10/1/2002 QLOGIC
 scsidev.inf Not Available
 SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
 DO_DEVICE&REV_5&167DDF5A&0&007F00
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&167DDF5A&0&011
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&167DDF5A&0&000
 QLogic Fibre Channel Adapter No SCSIADAPTER
 9.1.0.12 9/21/2005 QLogic oem7.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&167DDF5A&0&000
 QLogic Fibre Channel Adapter No SCSIADAPTER
 9.1.0.12 9/21/2005 QLogic oem7.inf Not
 Available
 PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
 REV_02\4&167C2C2F&0&0000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.1830 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
 REV_01\3&71811BB&0&00
 PCI bus No SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)

machine.inf Not Available
 ACPI\PNP0A08\6
 RAID Virtual Device No SYSTEM 5.2.3790.1830
 10/1/2002 American Megatrends, Inc.
 scsidev.inf Not Available
 SCSI\OTHER&VEN_RAID&PROD_DUMMYD
 EVICE&REV_0001\5&297181F&0&014000
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_SERVERAID-
 MR10M&REV_1.20\5&297181F&0&010000
 IBM ServeRAID-MR10M SAS/SATA Controller No
 SCSIADAPTER 2.21.0.64 2/15/2008 LSI
 Corp., oem28.inf Not Available
 PCI\VEN_1000&DEV_0060&SUBSYS_03791014&
 REV_04\4&1DD220B0&0&0000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.1830 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
 REV_01\3&19D1C260&0&00
 PCI bus No SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0A08\5
 Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client)
 No NET 3.7.6.0 7/2/2007
 Broadcom Corporation oem15.inf Not
 Available
 B06BDRVL2ND&PCI_164C14E4&SUBSYS_164C
 14E4&REV_12\6&33B9FAC3&0&20052B00
 Broadcom BCM5708C NetXtreme II GigE No
 SYSTEM 3.7.8.0 7/16/2007 Broadcom
 Corporation oem14.inf Not Available
 PCI\VEN_14E4&DEV_164C&SUBSYS_164C14E4
 &REV_12\5&27B57945&0&000000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.3959 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1166&DEV_0103&SUBSYS_00000000&
 REV_C3\4&394070D2&0&0000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.1830 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
 REV_01\3&2C8B7305&0&00
 PCI bus No SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0A08\4
 Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client)
 No NET 3.7.6.0 7/2/2007
 Broadcom Corporation oem15.inf Not
 Available

B06BDRVL2ND&PCI_164C14E4&SUBSYS_164C
 14E4&REV_12\6&7B02679&0&20052500
 Broadcom BCM5708C NetXtreme II GigE No
 SYSTEM 3.7.8.0 7/16/2007 Broadcom
 Corporation oem14.inf Not Available
 PCI\VEN_14E4&DEV_164C&SUBSYS_164C14E4
 &REV_12\5&22BE75FB&0&000000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.3959 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1166&DEV_0103&SUBSYS_00000000&
 REV_C3\4&98F6FDA&0&0000
 PCI standard PCI-to-PCI bridge No SYSTEM
 5.2.3790.1830 10/1/2002 (Standard system
 devices) machine.inf Not Available
 PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
 REV_01\3&3AA59A3&0&00
 PCI bus No SYSTEM 5.2.3790.1830
 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ACPI\PNP0A08\3
 Qlogic processor device No SYSTEM
 5.2.3790.1830 10/1/2002 QLOGIC
 scsidev.inf Not Available
 SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
 DO_DEVICE&REV_5&1CD887C9&0&007F00
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&1CD887C9&0&010
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&1CD887C9&0&001
 QLogic Fibre Channel Adapter No SCSIADAPTER
 9.1.0.12 9/21/2005 QLogic oem7.inf Not
 Available
 PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
 REV_02\4&15BF98E9&0&0100
 Qlogic processor device No SYSTEM
 5.2.3790.1830 10/1/2002 QLOGIC
 scsidev.inf Not Available
 SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
 DO_DEVICE&REV_5&43F601F&0&007F00
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available
 SCSI\DISK&VEN_IBM&PROD_1815____FAST
 T&REV_0914\5&43F601F&0&010
 Disk drive No DISKDRIVE 5.2.3790.1830
 10/1/2002 (Standard disk drives) disk.inf Not
 Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&43F601F&0&001
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.0.12 9/21/2005 QLogic oem7.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&15BF98E9&0&0000
PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3790.1830 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
REV_01\3&36CB97A3&0&00
PCI bus No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A08\2
Qlogic processor device No SYSTEM
5.2.3790.1830 10/1/2002 QLOGIC
scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
DO_DEVICE&REV_5&18E37560&0&007F00
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&028
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&026
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&024
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&022
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&020
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&011
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available

SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&18E37560&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.0.12 9/21/2005 QLogic oem7.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&1DD35BBD&0&0100
Qlogic processor device No SYSTEM
5.2.3790.1830 10/1/2002 QLOGIC
scsidev.inf Not Available
SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEU
DO_DEVICE&REV_5&317C9D0A&0&007F00
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&317C9D0A&0&011
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM&PROD_1815_____FAST
T&REV_0914\5&317C9D0A&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER
9.1.0.12 9/21/2005 QLogic oem7.inf Not
Available
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&
REV_02\4&1DD35BBD&0&0000
PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3790.1830 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
REV_01\3&2411E6FE&0&00
PCI bus No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A08\1
IBM SAS SES Device No SYSTEM 1.0.4.0
3/26/2008 IBM Corporation oem27.inf Not
Available
SCSI\ENCLOSURE&VEN_IBM&PROD_SAS_SES
-2_DEVICE&REV_01.0\5&22AD4189&0&001500
Disk drive No DISKDRIVE 5.2.3790.1830
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_IBM-
ESXS&PROD_ST973401SS&REV_B51D\5&22AD4189&0&0
00F00
LSI Adapter, SAS RAID-on-Chip, 8-port with 1078 -StorPort
No SCSIADAPTER 1.27.3.0 4/8/2008
LSI Corporation oem26.inf Not Available
PCI\VEN_1000&DEV_0062&SUBSYS_10001000&
REV_03\4&2BCCA3F5&0&0000
PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3790.3959 10/1/2002 (Standard system
devices) machine.inf Not Available

PCI\VEN_1014&DEV_0308&SUBSYS_00000000&
REV_01\3&11583659&0&00
PCI bus No SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A08\0
Memory Module No MEMORY
5.2.3790.1830 10/1/2002 Microsoft
memory.inf Not Available
ACPI\PNP0C80\0
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\23
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\22
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\21
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\20
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\19
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\18
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\17
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\16
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\15
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\14
Intel Processor No PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\13

Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\12		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\11		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\10		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\9		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\8		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\7		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\6		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\5		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\4		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\3		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\2		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\1		
Intel Processor	No	PROCESSOR
5.2.3790.1830	10/1/2002	Intel cpu.inf
Not Available		ACPI\GENUINEINTEL_-
_EM64T_FAMILY_6_MODEL_29\0		
ACPI Fixed Feature Button	No	SYSTEM
5.2.3790.1830	10/1/2002	(Standard system

devices)	machine.inf	Not Available
	ACPI\FIXEDBUTTON\2&DABA3FF&0	
Motherboard resources	No	SYSTEM
5.2.3790.3959	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0C31\1	
Motherboard resources	No	SYSTEM
5.2.3790.1830	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0C02\D	
System board	No	SYSTEM 5.2.3790.1830
10/1/2002	(Standard system devices)	
	machine.inf	Not Available
	ACPI\PNP0C01\1	
Motherboard resources	No	SYSTEM
5.2.3790.1830	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\IBM37D4\0	
Secondary IDE Channel	No	HDC
5.2.3790.1830	10/1/2002	(Standard IDE
ATA/ATAPI controllers)	mshdc.inf	Not Available
	PCIIDEM\DECHANNEL\4&37BDD80B&0&1	
CD-ROM Drive	No	CDROM 5.2.3790.1830
10/1/2002	(Standard CD-ROM drives)	
	cdrom.inf	Not Available IDE\CDROMHL-
DT-ST_RW\DVD_GCC-		
T10N_____1.00_____5&E769355&0&0.0.0		
Primary IDE Channel	No	HDC 5.2.3790.1830
10/1/2002	(Standard IDE ATA/ATAPI controllers)	
	mshdc.inf	Not Available
	PCIIDEM\DECHANNEL\4&37BDD80B&0&0	
Standard Dual Channel PCI IDE Controller	No	HDC
5.2.3790.1830	10/1/2002	(Standard IDE
ATA/ATAPI controllers)	mshdc.inf	Not Available
	PCI\VEN_8086&DEV_27DF&SUBSYS_03811014	
&REV_01\3&29D18217&0&F9		
Motherboard resources	No	SYSTEM
5.2.3790.1830	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PI0001\0	
Motherboard resources	No	SYSTEM
5.2.3790.1830	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0C02\3	
Numeric data processor	No	SYSTEM
5.2.3790.3959	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0C04\4&3695E41E&0	
System speaker	No	SYSTEM 5.2.3790.3959
10/1/2002	(Standard system devices)	
	machine.inf	Not Available

		ACPI\PNP0800\4&3695E41E&0
System CMOS/real time clock	No	SYSTEM
5.2.3790.3959	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0B00\4&3695E41E&0	
System timer	No	SYSTEM 5.2.3790.3959
10/1/2002	(Standard system devices)	
	machine.inf	Not Available
	ACPI\PNP0100\4&3695E41E&0	
Direct memory access controller	No	SYSTEM
5.2.3790.3959	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	ACPI\PNP0200\4&3695E41E&0	
Advanced programmable interrupt controller	No	SYSTEM 5.2.3790.3959
10/1/2002	(Standard system devices)	machine.inf
		Not Available
		ACPI\PNP0003\4&3695E41E&0
Communications Port No	PORTS	5.2.3790.1830
10/1/2002	(Standard port types)	msports.inf
		Not Available ACPI\PNP0501\1
PCI standard ISA bridge	No	SYSTEM
5.2.3790.3959	10/1/2002	(Standard system
devices)	machine.inf	Not Available
	PCI\VEN_8086&DEV_27B8&SUBSYS_00000000&	
REV_01\3&29D18217&0&F8		
Plug and Play Monitor	No	MONITOR
5.2.3790.1830	10/1/2002	(Standard monitor
types)	monitor.inf	Not Available
	DISPLAY\AVO0000\5&9D68E52&1&10000080&0	
1&00		
Default Monitor	No	MONITOR
5.2.3790.1830	10/1/2002	(Standard monitor
types)	monitor.inf	Not Available
	DISPLAY\DEFAULT_MONITOR\5&9D68E52&1&	
10000001&01&00		
Plug and Play Monitor	No	MONITOR
5.2.3790.1830	10/1/2002	(Standard monitor
types)	monitor.inf	Not Available
	DISPLAY\IBM029A\5&9D68E52&1&10000082&0	
1&00		
ATI ES1000	No	DISPLAY 8.24.3.0 4/5/2006
	ATI Technologies Inc.	oem5.inf Not
Available		
	PCI\VEN_1002&DEV_515E&SUBSYS_03191014&	
REV_02\4&18E510B5&0&00F0		
Intel(R) 82801 PCI Bridge - 244E	No	
SYSTEM 5.2.3790.3959	10/1/2002	Intel
	machine.inf	Not Available
	PCI\VEN_8086&DEV_244E&SUBSYS_00000000&	
REV_E1\3&29D18217&0&F0		
USB Root Hub	No	USB 5.2.3790.1830
10/1/2002	(Standard USB Host Controller)	
	usbport.inf	Not Available

```

USB\ROOT_HUB20\4&CEBC86B&0
Standard Enhanced PCI to USB Host Controller No
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf\Not
Available
PCI\VEN_8086&DEV_27CC&SUBSYS_03811014
&REV_01\3&29D18217&0&EF
USB Root Hub No USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf\Not Available
USB\ROOT_HUB\4&25C8C8F6&0
Standard Universal PCI to USB Host Controller No
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf\Not
Available
PCI\VEN_8086&DEV_27CB&SUBSYS_03811014
&REV_01\3&29D18217&0&EB
HID-compliant deviceNo HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
HID\VID_04B3&PID_4001&MI_02\7&1E0CE9C6
&0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
USB\VID_04B3&PID_4001&MI_02\6&2E1ACEF8
&0&0002
HID-compliant mouseNo MOUSE 5.2.3790.1830
10/1/2002 Microsoft msmouse.inf Not
Available
HID\VID_04B3&PID_4001&MI_01\7&6495D81&0
&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
USB\VID_04B3&PID_4001&MI_01\6&2E1ACEF8
&0&0001
HID Keyboard Device No KEYBOARD
5.2.3790.1830 10/1/2002 (Standard keyboards)
keyboard.inf Not Available
HID\VID_04B3&PID_4001&MI_00\7&2A209B43&
0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
USB\VID_04B3&PID_4001&MI_00\6&2E1ACEF8
&0&0000
USB Composite Device No USB
5.2.3790.1830 10/1/2002 (Standard USB Host
Controller)usb.inf Not Available
USB\VID_04B3&PID_4001\00145ECFBE68
USB Root Hub No USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller)

```

```

usbport.inf\Not Available
USB\ROOT_HUB\4&11F79082&0
Standard Universal PCI to USB Host Controller No
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf\Not
Available
PCI\VEN_8086&DEV_27CA&SUBSYS_03811014
&REV_01\3&29D18217&0&EA
USB Root Hub No USB 5.2.3790.1830
10/1/2002 (Standard USB Host Controller)
usbport.inf\Not Available
USB\ROOT_HUB\4&3ABE3947&0
Standard Universal PCI to USB Host Controller No
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf\Not
Available
PCI\VEN_8086&DEV_27C9&SUBSYS_03811014&
REV_01\3&29D18217&0&E9
HID-compliant deviceNo HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
HID\VID_0624&PID_0296&MI_01&COL03\7&D5
CFCD9&0&0002
HID-compliant consumer control device No
HIDCLASS 5.2.3790.1830
10/1/2002 Microsoft hidserv.inf\Not Available
HID\VID_0624&PID_0296&MI_01&COL02\7&D5
CFCD9&0&0001
HID-compliant mouseNo MOUSE 5.2.3790.1830
10/1/2002 Microsoft msmouse.inf Not
Available
HID\VID_0624&PID_0296&MI_01&COL01\7&D5
CFCD9&0&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
USB\VID_0624&PID_0296&MI_01\6&A56CC61&
0&0001
HID Keyboard Device No KEYBOARD
5.2.3790.1830 10/1/2002 (Standard keyboards)
keyboard.inf Not Available
HID\VID_0624&PID_0296&MI_00\7&5D31F65&0
&0000
USB Human Interface Device No HIDCLASS
5.2.3790.1830 10/1/2002 (Standard system
devices) input.inf Not Available
USB\VID_0624&PID_0296&MI_00\6&A56CC61&
0&0000
USB Composite Device No USB
5.2.3790.1830 10/1/2002 (Standard USB Host
Controller)usb.inf Not Available
USB\VID_0624&PID_0296\5&4203476&0&1
USB Root Hub No USB 5.2.3790.1830

```

```

10/1/2002 (Standard USB Host Controller)
usbport.inf\Not Available
USB\ROOT_HUB\4&389CEA38&0
Standard Universal PCI to USB Host Controller No
USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf\Not
Available
PCI\VEN_8086&DEV_27C8&SUBSYS_03811014&
REV_01\3&29D18217&0&E8
Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client)
No NET 3.7.6.0 7/2/2007
Broadcom Corporation oem15.inf Not
Available
B06BDRV\2ND&PCI_163914E4&SUBSYS_037C
1014&REV_01\5&38892308&0&20050200
Broadcom BCM5709C NetXtreme II GigE No
SYSTEM 3.7.8.0 7/16/2007 Broadcom
Corporation oem14.inf Not Available
PCI\VEN_14E4&DEV_1639&SUBSYS_037C1014
&REV_01\4&39B23F1D&0&01E0
Broadcom BCM5709C NetXtreme II GigE (NDIS VBD Client)
No NET 3.7.6.0 7/2/2007
Broadcom Corporation oem15.inf Not
Available
B06BDRV\2ND&PCI_163914E4&SUBSYS_037C
1014&REV_01\5&158780AB&0&20050200
Broadcom BCM5709C NetXtreme II GigE No
SYSTEM 3.7.8.0 7/16/2007 Broadcom
Corporation oem14.inf Not Available
PCI\VEN_14E4&DEV_1639&SUBSYS_037C1014
&REV_01\4&39B23F1D&0&00E0
PCI standard PCI-to-PCI bridge No SYSTEM
5.2.3790.3959 10/1/2002 (Standard system
devices) machine.inf Not Available
PCI\VEN_8086&DEV_27D0&SUBSYS_00000000
&REV_01\3&29D18217&0&E0
PCI bus No SYSTEM 5.2.3790.3959
10/1/2002 (Standard system devices)
machine.inf Not Available
ACPI\PNP0A08\E
Microsoft ACPI-Compliant System No
SYSTEM 5.2.3790.1830 10/1/2002
Microsoft acpi.inf Not Available
ACPI_HAL\PNP0C08\0
ACPI Multiprocessor x64-based PC No
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 Not Available Not Available
HTREE\ROOT\0

```

[Environment Variables]

```

Variable Value User Name
ClusterLog C:\WINDOWS\Cluster\cluster.log
<SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe
<SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
NUMBER_OF_PROCESSORS 24 <SYSTEM>
OS Windows_NT <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%System
Root%\System32\Wbem;c:\tools;c:\tools\util;c:\batfiles;C:\Progr
am Files (x86)\Microsoft SQL
Server\80\Tools\Binn\C:\Program Files\Microsoft SQL
Server\90\Tools\Binn\C:\Program Files (x86)\Microsoft SQL
Server\90\Tools\Binn\C:\Program Files (x86)\Microsoft SQL
Server\90\DTS\Binn\C:\Program Files (x86)\Microsoft SQL
Server\90\Tools\Binn\VSShell\Common7\IDE\C:\Program Files
(x86)\Microsoft Visual Studio
8\Common7\IDE\PrivateAssemblies;c:\Program Files\Microsoft
SQL Server\90\DTS\Binn\ <SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE AMD64
<SYSTEM>
PROCESSOR_IDENTIFIER EM64T Family 6 Model 29
Stepping 1, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 1d01 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
lib C:\Program Files\SQLXML 4.0\bin\
<SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp

```

```

IBMSERV4\Administrator
TMP %USERPROFILE%\Local Settings\Temp
IBMSERV4\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
x: \\fsserv\rayDisk Persistent Connection
IBMSERV4\Administrator
y: \\fsserv\edrive Disk Persistent
Connection IBMSERV4\Administrator

[Running Tasks]

Name Path Process ID Priority Min Working Set
Max Working Set Start Time Version Size
File Date
system idle process Not Available 0 0
Not Available Not Available Not
Available Not Available Not Available Not
system Not Available 4 8 0
1380 Not Available Not Available
smss.exe Not Available 360 11 200
1380 8/28/2008 2:43 PM Not Available
Not Available Not Available
csrss.exe Not Available 408 13 Not
Available Not Available 8/28/2008 2:43 PM Not
Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe
436 13 200 1380
8/28/2008 2:43 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 922.00 KB (944,128 bytes)
5/25/2007 12:13 PM
services.exe c:\windows\system32\services.exe
484 9 200 1380
8/28/2008 2:43 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 219.00 KB (224,256 bytes)
3/25/2005 8:00 AM
lsass.exe c:\windows\system32\lsass.exe 496 9
200 1380 8/28/2008 2:43 PM
5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
14.00 KB (14,336 bytes) 3/25/2005 8:00 AM
svchost.exe c:\windows\system32\svchost.exe

```

```

652 8 200 1380
8/28/2008 2:43 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 25.00 KB (25,600 bytes)
5/25/2007 12:13 PM
svchost.exe Not Available 740 8
Not Available Not Available
8/28/2008 2:43 PM Not Available Not
Available Not Available
svchost.exe Not Available 808 8
Not Available Not Available
8/28/2008 2:43 PM Not Available Not
Available Not Available
svchost.exe c:\windows\system32\svchost.exe
836 8 200 1380
8/28/2008 2:43 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 25.00 KB (25,600 bytes)
5/25/2007 12:13 PM
msdtc.exe Not Available 932 8 Not
Available Not Available 8/28/2008 2:43 PM Not
Available Not Available Not Available
svchost.exe Not Available 1072 8
Not Available Not Available
8/28/2008 2:43 PM Not Available Not
Available Not Available
explorer.exe c:\windows\explorer.exe 1620
8 200 1380 8/28/2008 2:43 PM
6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
1.30 MB (1,364,480 bytes) 5/25/2007 12:14
PM
svchost.exe c:\windows\system32\svchost.exe
1676 8 200 1380
8/28/2008 2:43 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 25.00 KB (25,600 bytes)
5/25/2007 12:13 PM
wmiprvse.exe Not Available 1936 8
Not Available Not Available
8/28/2008 2:43 PM Not Available Not
Available Not Available
helpctr.exe c:\windows\pchealth\helpctr\binaries\helpctr.exe
792 8 200 1380
8/28/2008 2:51 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 1.30 MB (1,363,456 bytes)
5/25/2007 9:06 AM
wmiprvse.exe Not Available 1068 8
Not Available Not Available
8/28/2008 2:51 PM Not Available Not
Available Not Available
helpsvc.exe c:\windows\pchealth\helpctr\binaries\helpsvc.exe
1164 8 200 1380
8/28/2008 2:51 PM 5.2.3790.3959
(srv03_sp2_rtm.070216-1710) 1.52 MB (1,591,296 bytes)
5/25/2007 9:06 AM

```

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer
winlogon	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	922.00 KB (944,128 bytes)	5/25/2007 12:13	Microsoft Corporation
ntdll	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.20 MB (1,254,400 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
kernel32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.43 MB (1,503,232 bytes)	5/25/2007 12:13	Microsoft Corporation
advapi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.00 MB (1,051,648 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
rpcrt4	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.58 MB (1,653,248 bytes)	5/25/2007 12:13	Microsoft Corporation
secur32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	120.00 KB (122,880 bytes)	5/25/2007 12:13	Microsoft Corporation
crypt32	5.131.3790.3959 (srv03_sp2_rtm.070216-1710)	1.36 MB (1,429,504 bytes)	5/25/2007 12:14	Microsoft Corporation
msvcrt	7.0.3790.3959 (srv03_sp2_rtm.070216-1710)	508.00 KB (520,192 bytes)	5/25/2007 12:13	Microsoft Corporation
user32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.04 MB (1,086,976 bytes)	5/25/2007 12:13	Microsoft Corporation
gdi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	603.50 KB (617,984 bytes)	5/25/2007 12:13	Microsoft Corporation
msasn1	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	152.50 KB (156,160 bytes)	5/25/2007 12:13	Microsoft Corporation
nddeapi	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	25.00 KB (25,600 bytes)	5/25/2007 12:13	Microsoft Corporation
profmap	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	36.00 KB (36,864 bytes)	5/25/2007 12:13	Microsoft Corporation

netapi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	589.00 KB (603,136 bytes)	5/25/2007 12:13	Microsoft Corporation
userenv	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.02 MB (1,071,104 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
psapi	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	29.00 KB (29,696 bytes)	5/25/2007 12:13	Microsoft Corporation
regapi	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	108.50 KB (111,104 bytes)	5/25/2007 12:13	Microsoft Corporation
setupapi	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.45 MB (1,524,224 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
version	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB (28,672 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
winsta	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	89.00 KB (91,136 bytes)	5/25/2007 12:13	Microsoft Corporation
ws2_32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	174.50 KB (178,688 bytes)	5/25/2007 12:13	Microsoft Corporation
ws2help	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	30.50 KB (31,232 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
msgina	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.14 MB (1,193,472 bytes)	5/25/2007 12:13	Microsoft Corporation
shsvcs	6.00.3790.3959 (srv03_sp2_rtm.070216-1710)	194.00 KB (198,656 bytes)	5/25/2007 12:13	Microsoft Corporation
shlwapi	6.00.3790.3959 (srv03_sp2_rtm.070216-1710)	607.00 KB (621,568 bytes)	5/25/2007 12:13	Microsoft Corporation
sfc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	6.00 KB (6,144 bytes)	3/25/2005 8:00 AM	Microsoft Corporation

sfc_os	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	183.50 KB (187,904 bytes)	5/25/2007 12:13	Microsoft Corporation
wintrust	5.131.3790.3959 (srv03_sp2_rtm.070216-1710)	297.50 KB (304,640 bytes)	5/25/2007 12:13	Microsoft Corporation
imagehlp	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	57.50 KB (58,880 bytes)	3/25/2005 8:00 AM	Microsoft Corporation
ole32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	2.50 MB (2,622,976 bytes)	5/25/2007 12:13	Microsoft Corporation
comctl32	6.0 (srv03_sp2_rtm.070216-1710)	1.51 MB (1,584,640 bytes)	2/18/2007 10:24 AM	Microsoft Corporation
winscard	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	229.50 KB (235,008 bytes)	5/25/2007 12:13	Microsoft Corporation
wtsapi32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	29.00 KB (29,696 bytes)	5/25/2007 12:13	Microsoft Corporation
winmm	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	303.50 KB (310,784 bytes)	5/25/2007 12:13	Microsoft Corporation
shell32	6.00.3790.3959 (srv03_sp2_rtm.070216-1710)	10.02 MB (10,505,728 bytes)	5/25/2007 12:13	Microsoft Corporation
sxs	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	1.89 MB (1,977,856 bytes)	5/25/2007 12:13	Microsoft Corporation
rsaenh	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	298.84 KB (306,008 bytes)	5/25/2007 12:13	Microsoft Corporation
wldap32	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)	390.50 KB (399,872 bytes)	5/25/2007 12:13	Microsoft Corporation
ati2evxx	6.14.10.4131	133.00 KB (136,192 bytes)	4/5/2006 9:57 PM	ATI Technologies Inc.
cscdll	5.2.3790.3959 (srv03_sp2_rtm.070216-1710)			Microsoft Corporation

151.50 KB (155,136 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\csddl.dll
 dimsntfy 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 28.00 KB (28,672 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\dimsntfy.dll
 wlnotify 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 148.50 KB (152,064 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\wlnotify.dll
 winspool 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 247.00 KB (252,928 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\winspool.drv
 mpr 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 115.00 KB (117,760 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\mpr.dll
 oleaut32 5.2.3790.3959 1.07 MB (1,121,792 bytes)
 3/25/2005 8:00 AM Microsoft Corporation
 c:\windows\system32\oleaut32.dll
 comctl32 5.82 (srv03_sp2_rtm.070216-1710) 935.00
 KB (957,440 bytes) 2/18/2007 10:24 AM Microsoft
 Corporation
 c:\windows\winsxs\amd64_microsoft.windows.comm
 on-controls_6595b64144ccfd5_82.3790.3959_x-
 ww_ab06deb0\comctl32.dll
 uxtheme 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
 494.50 KB (506,368 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\uxtheme.dll
 samlib 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 69.50 KB (71,168 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\samlib.dll
 cscuri 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 441.00 KB (451,584 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\cscuri.dll
 mprapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 154.50 KB (158,208 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\mprapi.dll
 activeds 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 348.50 KB (356,864 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\activeds.dll
 adslsdp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 240.50 KB (246,272 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\adslsdp.dll
 credui 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 202.00 KB (206,848 bytes) 5/25/2007 12:14

PM Microsoft Corporation
 c:\windows\system32\credui.dll
 atl 3.05.2284 96.50 KB (98,816 bytes)
 3/25/2005 8:00 AM Microsoft Corporation
 c:\windows\system32\atl.dll
 rtutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 66.00 KB (67,584 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\rtutils.dll
 ntlanman 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 72.00 KB (73,728 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\ntlanman.dll
 netui0 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 130.00 KB (133,120 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\netui0.dll
 netui1 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 338.50 KB (346,624 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\netui1.dll
 drprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 24.00 KB (24,576 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\drprov.dll
 davclnt 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 39.50 KB (40,448 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\davclnt.dll
 mprui 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 67.50 KB (69,120 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\mprui.dll
 netui2 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 542.00 KB (555,008 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\netui2.dll
 comdlg32 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
 447.00 KB (457,728 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\comdlg32.dll
 netmsg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 179.00 KB (183,296 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\netmsg.dll
 ntmarta 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 222.50 KB (227,840 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\ntmarta.dll
 clbcatq 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)
 862.50 KB (883,200 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\clbcatq.dll
 comres 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)

779.50 KB (798,208 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\comres.dll
 xpsp2res 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 2.77 MB (2,899,456 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\xpsp2res.dll
 services 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 219.00 KB (224,256 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\services.exe
 sceesrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 595.00 KB (609,280 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\sceesrv.dll
 authz 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 171.00 KB (175,104 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\authz.dll
 umpnpgm 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 205.00 KB (209,920 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\umpnpgm.dll
 ncobjapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 77.50 KB (79,360 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\ncobjapi.dll
 msvcp60 7.0.3790.1830 (srv03_sp1_rtm.050324-1447)
 919.50 KB (941,568 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\msvcp60.dll
 eventlog 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 127.50 KB (130,560 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\eventlog.dll
 lsass 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 14.00 KB (14,336 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\lsass.exe
 lsasrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 1.49 MB (1,566,720 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\lsasrv.dll
 samsrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 1.01 MB (1,059,328 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\samsrv.dll
 cryptdll 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 47.00 KB (48,128 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\cryptdll.dll
 dnsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 298.50 KB (305,664 bytes) 5/25/2007 12:14
 PM Microsoft Corporation

ntdsapi c:\windows\system32\ntdsapi.dll
 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 127.50 KB (130,560 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 msprivs c:\windows\system32\ntdsapi.dll
 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 47.50 KB (48,640 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\msprivs.dll
 kerberos 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 706.00 KB (722,944 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\kerberos.dll
 msv1_0 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 253.00 KB (259,072 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\msv1_0.dll
 iphlpapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 178.50 KB (182,784 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\iphlpapi.dll
 netlogon 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 665.50 KB (681,472 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\netlogon.dll
 w32time 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 398.00 KB (407,552 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\w32time.dll
 schannel 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 248.00 KB (253,952 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\schannel.dll
 wdigest 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 131.00 KB (134,144 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\wdigest.dll
 rassfm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 36.00 KB (36,864 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\rassfm.dll
 kdcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 410.00 KB (419,840 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\kdcsvc.dll
 ntdsa 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 2.83 MB (2,967,040 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\ntdsa.dll
 ntdsatq 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 51.00 KB (52,224 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\ntdsatq.dll
 mswsock 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)

480.50 KB (492,032 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\mswsock.dll
 esent 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 2.26 MB (2,367,488 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\esent.dll
 scecli 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 308.00 KB (315,392 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\scecli.dll
 ws03res 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 794.00 KB (813,056 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\ws03res.dll
 pstorsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 36.00 KB (36,864 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\pstorsvc.dll
 psbase 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 124.00 KB (126,976 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\psbase.dll
 hnetcfg 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 568.00 KB (581,632 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\hnetcfg.dll
 wshtcpip 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 29.00 KB (29,696 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\wshtcpip.dll
 dssenh 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 231.34 KB (236,888 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\dssenh.dll
 svchost 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 25.00 KB (25,600 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\svchost.exe
 rpcss 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 819.00 KB (838,656 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\rpcss.dll
 schedsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 308.00 KB (315,392 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\schedsvc.dll
 msidle 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 9.00 KB (9,216 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\msidle.dll
 wkssvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 221.00 KB (226,304 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation

c:\windows\system32\wkssvc.dll
 wiarpc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 57.00 KB (58,368 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\wiarpc.dll
 aelupsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 31.50 KB (32,256 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\aelupsvc.dll
 apphelp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 241.00 KB (246,784 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\apphelp.dll
 dmserver 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 36.50 KB (37,376 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\dmserver.dll
 es 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)
 357.00 KB (365,568 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\es.dll
 pchsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 75.50 KB (77,312 bytes) 5/25/2007 9:06 AM
 Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\pchsvc.dll
 hidserv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 24.00 KB (24,576 bytes) 5/24/2007 2:11 PM
 Microsoft Corporation
 c:\windows\system32\hidserv.dll
 hid 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 33.00 KB (33,792 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\hid.dll
 srvsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 156.50 KB (160,256 bytes) 3/25/2005 8:00 AM
 Microsoft Corporation
 c:\windows\system32\srvsvc.dll
 seclogon 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 27.50 KB (28,160 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\seclogon.dll
 trkwks 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 177.50 KB (181,760 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\trkwks.dll
 sens 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 63.50 KB (65,024 bytes) 5/25/2007 12:13
 PM Microsoft Corporation
 c:\windows\system32\sens.dll
 wmisvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
 227.50 KB (232,960 bytes) 5/25/2007 12:14
 PM Microsoft Corporation
 c:\windows\system32\wbem\wmisvc.dll

vssapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.26 MB (1,320,960 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\vssapi.dll
comsvcs 2001.12.4720.3959 (srv03_sp2_rtm.070216-1710)
2.13 MB (2,234,880 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\comsvcs.dll
netman 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
454.50 KB (465,408 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\netman.dll
netshell 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
2.33 MB (2,438,656 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\netshell.dll
clusapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
127.00 KB (130,048 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\clusapi.dll
rasapi32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
410.00 KB (419,840 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\rasapi32.dll
rasman 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
95.50 KB (97,792 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\rasman.dll
tapi32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
332.50 KB (340,480 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\tapi32.dll
wzcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
644.50 KB (659,968 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\wzcsvc.dll
wmi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
5.50 KB (5,632 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\wmi.dll
dhcpcsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
220.50 KB (225,792 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\dhcpcsvc.dll
wininet 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
1.14 MB (1,190,912 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\wininet.dll
wzcsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
50.50 KB (51,712 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\wzcsapi.dll
wbemcomn 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
518.00 KB (530,432 bytes) 5/25/2007 12:14

PM Microsoft Corporation
c:\windows\system32\wbem\wbemcomn.dll
wbemcore 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.22 MB (1,282,560 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
626.50 KB (641,536 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\esscli.dll
fastprox 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
866.50 KB (887,296 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\fastprox.dll
wbemsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
58.00 KB (59,392 bytes) 5/25/2007 9:03 AM
Microsoft Corporation
c:\windows\system32\wbem\wbemsvc.dll
wmiutils 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
171.00 KB (175,104 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\wmiutils.dll
reprdrvfs 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
353.50 KB (361,984 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\reprdrvfs.dll
wmiprvsd 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
743.00 KB (760,832 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
532.50 KB (545,280 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\wbem\wbemess.dll
rasdlg 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
859.50 KB (880,128 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\rasdlg.dll
ncprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
73.00 KB (74,752 bytes) 5/25/2007 9:03 AM
Microsoft Corporation
c:\windows\system32\wbem\ncprov.dll
rasadhlp 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
12.00 KB (12,288 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\rasadhlp.dll
wbemcons 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
65.50 KB (67,072 bytes) 5/25/2007 9:03 AM
Microsoft Corporation
c:\windows\system32\wbem\wbemcons.dll
ntlsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
11.00 KB (11,264 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\ntlsapi.dll

explorer 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
1.30 MB (1,364,480 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\explorer.exe
browseui 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
1.53 MB (1,605,120 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\browseui.dll
shdocvw 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
2.33 MB (2,438,144 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\shdocvw.dll
cryptui 5.131.3790.3959 (srv03_sp2_rtm.070216-1710)
705.50 KB (722,432 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\cryptui.dll
themeui 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
531.50 KB (544,256 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\themeui.dll
msimg32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
6.50 KB (6,656 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\msimg32.dll
actxprxy 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
220.50 KB (225,792 bytes) 5/25/2007 12:14
PM Microsoft Corporation
c:\windows\system32\actxprxy.dll
linkinfo 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
31.00 KB (31,744 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\linkinfo.dll
ntshrui 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
184.00 KB (188,416 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\ntshrui.dll
urlmon 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
1.04 MB (1,088,000 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\urlmon.dll
webcheck 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
438.50 KB (449,024 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\webcheck.dll
wsock32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
24.50 KB (25,088 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\wsock32.dll
stobject 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
142.50 KB (145,920 bytes) 5/25/2007 12:13
PM Microsoft Corporation
c:\windows\system32\stobject.dll
batmeter 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
41.50 KB (42,496 bytes) 5/25/2007 12:14

PM Microsoft Corporation
c:\windows\system32\batmeter.dll

powrprof 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
32.50 KB (33,280 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\powrprof.dll

shdoclc 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
589.50 KB (603,648 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\shdoclc.dll

termsrv 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
355.50 KB (364,032 bytes) 5/25/2007 12:13

PM Microsoft Corporation
c:\windows\system32\termsrv.dll

icaapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
27.50 KB (28,160 bytes) 5/25/2007 9:03 AM
Microsoft Corporation
c:\windows\system32\icaapi.dll

mstlsapi 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
188.00 KB (192,512 bytes) 5/25/2007 12:13

PM Microsoft Corporation
c:\windows\system32\mstlsapi.dll

rdpwsx 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
170.13 KB (174,216 bytes) 5/25/2007 12:13

PM Microsoft Corporation
c:\windows\system32\rdpwsx.dll

helpctr 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.30 MB (1,363,456 bytes) 5/25/2007 9:06 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\helpctr.exe

hcappres 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
7.50 KB (7,680 bytes) 5/25/2007 9:06 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\hcappres.dll

itss 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
208.50 KB (213,504 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\itss.dll

msxml3 8.80.1185.0 2.04 MB (2,144,256 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\msxml3.dll

pchshell 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
155.00 KB (158,720 bytes) 5/25/2007 9:06 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\pchshell.dll

mlang 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
686.00 KB (702,464 bytes) 5/25/2007 12:13

PM Microsoft Corporation
c:\windows\system32\mlang.dll

mshtml 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
5.72 MB (5,999,616 bytes) 3/25/2005 8:00 AM

Microsoft Corporation
c:\windows\system32\mshhtml.dll

mshls31 3.10.349.0 357.00 KB (365,568 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\mshls31.dll

msimtf 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
380.50 KB (389,632 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\msimtf.dll

msctf 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
617.50 KB (632,320 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\msctf.dll

jscrip 5.6.0.8832 976.00 KB (999,424 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\jscrip.dll

imm32 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
208.00 KB (212,992 bytes) 5/25/2007 12:13

PM Microsoft Corporation
c:\windows\system32\imm32.dll

mshtml 6.00.3790.3959 (srv03_sp2_rtm.070216-1710)
905.50 KB (927,232 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\mshtml.dll

vbscript 5.6.0.8832 647.00 KB (662,528 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\vbscript.dll

msinfo 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
635.50 KB (650,752 bytes) 5/25/2007 9:06 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\msinfo.dll

mfc42u 6.50.9146.0 1.39 MB (1,460,992 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\mfc42u.dll

riched32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
7.00 KB (7,168 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\riched32.dll

riched20 5.31.23.1225 1.11 MB (1,160,192 bytes)
3/25/2005 8:00 AM Microsoft Corporation
c:\windows\system32\riched20.dll

wbemprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
38.00 KB (38,912 bytes) 5/25/2007 9:03 AM
Microsoft Corporation
c:\windows\system32\wbem\wbemprox.dll

browselc 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
63.00 KB (64,512 bytes) 3/25/2005 8:00 AM
Microsoft Corporation
c:\windows\system32\browselc.dll

helpsvc 5.2.3790.3959 (srv03_sp2_rtm.070216-1710)
1.52 MB (1,591,296 bytes) 5/25/2007 9:06 AM
Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\helpsvc.exe

[Services]

Display Name	Name	State	Start Mode	Service
Type	Path	Error Control	Start Name	Tag ID
Application Experience	Lookup Service	Running	Auto	AeLookupSvc
		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\framework64\v2.0.50727\aspnet_state.exe	Normal	NT	
AUTHORITY\NetworkService	0			
Ati HotKey Poller	Ati HotKey Poller	Stopped	Manual	
	Own Process			
	c:\windows\system32\ati2evxx.exe	Normal		
	LocalSystem	0		
Windows Audio	AudioSrv	Stopped	Disabled	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	Stopped	Manual	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\v2.0.50727\mscorsvw.exe	Ignore	LocalSystem	0
.NET Runtime Optimization Service v2.0.50727_x64				

clr_optimization_v2.0.50727_64	Stopped	Manual		
Own Process				
c:\windows\microsoft.net\framework64\v2.0.50727\mscorlib.exe	Ignore	LocalSystem	0	
COM+ System Application	COMSysApp	Stopped		
Manual Own Process				
c:\windows\system32\dlhhost.exe				
/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}	Normal			
LocalSystem	0			
Cryptographic Services	CryptSvc	Stopped	Manual	
Share Process				
c:\windows\system32\svchost.exe -k netsvcs	Normal			
LocalSystem	0			
DCOM Server Process Launcher	DcomLaunch	Running		
Auto Share Process				
c:\windows\system32\svchost.exe -k dcomlaunch	Normal	LocalSystem	0	
Distributed File System	Dfs	Stopped	Manual	
Own Process				
c:\windows\system32\dfssvc.exe	Normal	LocalSystem	0	
DHCP Client	Dhcp	Stopped	Manual	Share
Process				
c:\windows\system32\svchost.exe -k networkservice	Normal	NT AUTHORITY\NetworkService	0	
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	Share
Process				
c:\windows\system32\dmadmin.exe /com	Normal	LocalSystem	0	
Logical Disk Managerdmsrvr	Running	Auto	Share	
Process				
c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0	
DNS Client	Dnscache	Stopped	Manual	Share
Process				
c:\windows\system32\svchost.exe -k networkservice	Normal	NT AUTHORITY\NetworkService	0	
Error Reporting Service	ERSvc	Stopped	Manual	
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	
HID Input Service	HidServ	Running	Auto	Share
Process				
c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0	

HTTP SSLHTTPFilter	Stopped	Manual	Share	
Process				
c:\windows\system32\lsass.exe	Normal	LocalSystem	0	
IAS Jet Database Access	IASJet	Stopped	Manual	
Share Process				
c:\windows\system32\svchost.exe -k iasjet	Normal	LocalSystem	0	
InstallDriver Table Manager	IDriverT	Stopped	Manual	
Own Process				
"c:\program files (x86)\common files\installshield\driver\1050\intel32\idrivert.exe"	Ignore	LocalSystem	0	
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled	Own Process
LocalSystem	0			
c:\windows\system32\imapi.exe	Normal	LocalSystem	0	
Intersite Messaging	IsmServ	Stopped	Disabled	Own
Process				
c:\windows\system32\ismserv.exe	Normal	LocalSystem	0	
Kerberos Key Distribution Center	kdc	Stopped		
Disabled Share Process				
c:\windows\system32\lsass.exe	Normal	LocalSystem	0	
Server	lanmanserver	Running	Auto	Share
Process				
c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0	
Workstation	lanmanworkstation	Running	Auto	
Share Process				
c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0	
License Logging	LicenseService	Stopped	Disabled	Own Process
Normal				
NT AUTHORITY\NetworkService	0			
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	
Share Process				
c:\windows\system32\svchost.exe -k localservice	Normal	NT AUTHORITY\LocalService	0	
MRMonitor	MegaMonitorSrv	Stopped	Manual	
Own Process				
"c:\program files (x86)\megaraid storage manager\megamonitor\mrmonitor.exe"	Normal	LocalSystem	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	

SQL Server FullText Search (MSSQLSERVER)	msftesql	Stopped	Disabled	Own Process
"c:\program files\microsoft sql server\mssql.1\mssql\bin\msftesql.exe" -s:mssql.1 -f:mssqlserver	Normal	LocalSystem	0	
Windows Installer	MSIServer	Stopped	Manual	Share
Process				
c:\windows\system32\msiexec.exe /v	Normal	LocalSystem	0	
MSMFramework	MSMFramework	Stopped	Manual	
Own Process				
"c:\program files (x86)\megaraid storage manager\framework\vivaldiframework.exe"	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	Manual	
Own Process				
"c:\program files\microsoft sql server\90\shared\sqladhlp90.exe"	Normal	NT AUTHORITY\NetworkService	0	
SQL Server Analysis Services (MSSQLSERVER)	MSSQLServerOLAPService	Stopped	Manual	
Own Process				
"c:\program files\microsoft sql server\mssql.2\olap\bin\msmdsrvr.exe" -s "c:\program files\microsoft sql server\mssql.2\olap\config"	Normal	LocalSystem	0	
Network DDE	NetDDE	Stopped	Disabled	Share
Process				
c:\windows\system32\netdde.exe	Normal	LocalSystem	0	
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	
Share Process				
c:\windows\system32\netdde.exe	Normal	LocalSystem	0	
Net Logon	Netlogon	Stopped	Manual	Share Process
c:\windows\system32\lsass.exe	Normal	LocalSystem	0	
Network Connections	Netman	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	NtFrs	Stopped	Manual	Own
Process				
c:\windows\system32\ntfrs.exe	Ignore	LocalSystem	0	
NT LM Security Support Provider	NtLmSsp	Stopped		
Manual Share Process				
c:\windows\system32\lsass.exe	Normal	LocalSystem	0	

Removable Storage Process	NtmsSvc	Stopped	Manual	Share	Normal	LocalSystem	0
Office Source Engine Process	ose	Stopped	Manual	Own	Normal	LocalSystem	0
Plug and Play Process	PlugPlay	Running	Auto	Share	Normal	LocalSystem	0
IPSEC Services	PolicyAgent	Stopped	Manual	Share Process	Normal	LocalSystem	0
Protected Storage	ProtectedStorage	Running	Auto	Share Process	Normal	LocalSystem	0
Remote Access Manual Process	Auto Connection Manager	RasAuto	Stopped	Share Process	Normal	LocalSystem	0
Remote Access Manual Process	Connection Manager	RasMan	Stopped	Share Process	Normal	LocalSystem	0
Remote Desktop Help Session Manager	RDSessMgr	Stopped	Manual	Own Process	Normal	LocalSystem	0
Routing and Remote Access Disabled Process	RemoteAccess	Stopped	Share Process	Normal	LocalSystem	0	
Remote Registry Share Process	RemoteRegistry	Running	Auto	Normal	LocalSystem	0	
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual	Own Process	Normal	LocalSystem	0
Remote Procedure Call (RPC) Share Process	RpcSs	Running	Auto	Normal	LocalSystem	0	
Resultant Set of Policy Provider	RSOPProv	Stopped	Manual	Share Process	Normal	LocalSystem	0
Special Administration Console Helper	sacsrv	Stopped	Manual	Share Process	Normal	LocalSystem	0
Security Accounts Manager	SamSs	Running	Auto	Share Process	Normal	LocalSystem	0

Smart Card Process	SCardSvr	Stopped	Manual	Share	Normal	LocalSystem	0
Task Scheduler Process	Schedule	Running	Auto	Share	Normal	LocalSystem	0
Secondary Logon Process	seclogon	Running	Auto	Share	Normal	LocalSystem	0
System Event Notification	SENS	Running	Auto	Share Process	Normal	LocalSystem	0
Windows Firewall/Internet Connection Sharing (ICS) SharedAccess	Stopped	Disabled	Share	Normal	LocalSystem	0	
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share Process	Normal	LocalSystem	0
Print Spooler Process	Spooler	Stopped	Manual	Own	Normal	LocalSystem	0
SQL Server Browser	SQLBrowser	Stopped	Disabled	Own Process	Normal	LocalSystem	0
SQL Server Agent (MSSQLSERVER)	SQLSERVERAGENT	Stopped	Manual	Own Process	Normal	LocalSystem	0
SQL Server VSS Writer	SQLWriter	Stopped	Disabled	Own Process	Normal	LocalSystem	0
Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled	Share Process	Normal	LocalSystem	0
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual	Own Process	Normal	LocalSystem	0
Performance Logs and Alerts	SysmonLog	Stopped	Auto	Own Process	Normal	LocalSystem	0
Telephony	TapiSrv	Stopped	Manual	Share Process	Normal	LocalSystem	0
Terminal Services	TermService	Running	Manual	Share Process	Normal	LocalSystem	0

Themes	Themes	Stopped	Disabled	Share Process	Normal	LocalSystem	0
Telnet	TlntSvr	Stopped	Disabled	Own Process	Normal	LocalSystem	0
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	Share Process	Normal	LocalSystem	0
Distributed Link Tracking Client	TrkWks	Running	Auto	Share Process	Normal	LocalSystem	0
Terminal Services Session Directory	Tssdis	Stopped	Disabled	Own Process	Normal	LocalSystem	0
Windows User Mode Driver Framework	UMWdf	Stopped	Manual	Own Process	Normal	LocalSystem	0
Uninterruptible Power Supply	UPS	Stopped	Manual	Own Process	Normal	LocalSystem	0
Virtual Disk Service	vds	Stopped	Manual	Own Process	Normal	LocalSystem	0
Volume Shadow Copy	VSS	Stopped	Manual	Own Process	Normal	LocalSystem	0
Windows Time	W32Time	Running	Auto	Share Process	Normal	LocalSystem	0
WebClient	WebClient	Stopped	Disabled	Share Process	Normal	LocalSystem	0
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share Process	Normal	LocalSystem	0
Windows Management Instrumentation	wimgmt	Running	Auto	Share Process	Normal	LocalSystem	0
Portable Media Serial Number Service	WmdmPmSN	Running	Manual	Share Process	Normal	LocalSystem	0

```

Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Windows Management Instrumentation Driver Extensions
Wmi Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
WMI Performance Adapter WmiApSrv Stopped
Manual Own Process
c:\windows\system32\wbem\wmiapsrv.exe Normal
LocalSystem 0
Automatic Updates wuauerv Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Wireless Configuration WZCSVC Stopped Manual
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
Accessories\Entertainment	Default	User:Accessories\Entertainment	Default
User:Accessories\Entertainment	Default	User:Accessories\Entertainment	Default
Startup	Default	User:Startup	Default
7-Zip	All Users:7-Zip	All Users	
Accessories	All Users:Accessories	All Users	
Accessories\Accessibility	All	User:Accessories\Accessibility	All
Users:Accessories\Accessibility	All	User:Accessories\Accessibility	All
Accessories\Communications	All	User:Accessories\Communications	All
Users:Accessories\Communications	All	User:Accessories\Communications	All
Accessories\Entertainment	All	User:Accessories\Entertainment	All
Users:Accessories\Entertainment	All	User:Accessories\Entertainment	All
Accessories\System Tools	All Users:Accessories\System	User:Accessories\System	All
Tools	All Users	User:Tools	All
Administrative Tools	All Users:Administrative Tools	All	
Users	All Users	User:Users	All
Broadcom	All Users:Broadcom	All Users	
MegaRAID Storage Manager	All Users:MegaRAID Storage	All Users	
Manager	All Users	User:Manager	All
Microsoft SQL Server 2005	All Users:Microsoft SQL	All Users	
Server 2005	All Users	User:Microsoft SQL Server 2005	All
Microsoft SQL Server 2005\Analysis Services	All	User:Microsoft SQL Server 2005\Analysis Services	All
Users:Microsoft SQL Server 2005\Analysis Services	All	User:Users	All
Microsoft SQL Server 2005\Configuration Tools	All	User: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
WinAMD64 Application Exerciser All
Users:WinAMD64 Application Exerciser 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 IBMSERV4\Administrator:Accessories
IBMSERV4\Administrator
Accessories\Accessibility
IBMSERV4\Administrator:Accessories\Accessibility
IBMSERV4\Administrator
Accessories\Entertainment
IBMSERV4\Administrator:Accessories\Entertainmen
t IBMSERV4\Administrator
Administrative Tools IBMSERV4\Administrator:Administrative
Tools IBMSERV4\Administrator
Startup IBMSERV4\Administrator:Startup
IBMSERV4\Administrator

```

[Startup Programs]

Program	Command	User Name	Location	
desktop	desktop.ini	NT AUTHORITY\SYSTEM		Startup
desktop	desktop.ini	IBMSERV4\Administrator		Startup
desktop	desktop.ini	.DEFAULT		Startup
desktop	desktop.ini	All Users	Common	Startup
Shortcut to Users	synchtime	c:\batfiles\syncht~1.bat		All

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /midi
Sound	Not Available
Media Clip	Not Available
WordPad Document	"%programfiles%\windows
	nt\accessories\wordpad.exe"
Bitmap Image	mspaint.exe

[Windows Error Reporting]

Time	Type	Details
------	------	---------

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
[Summary]

Item	Value
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			221 KB
Corporation	2/17/2007 12:02:52 AM			Microsoft
advpack.dll	6.0.3790.3959			146 KB
Corporation	2/17/2007 12:03:10 AM			Microsoft
asctrls.ocx	6.0.3790.1830	147 KB	3/25/2005 8:00:00 AM	
Corporation	C:\WINDOWS\system32			Microsoft
browseic.dll	6.0.3790.1830			63 KB

3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation			
browseui.dll	6.0.3790.3959	1,568 KB	2/17/2007 12:05:24 AM	C:\WINDOWS\system32	Microsoft Corporation
cdview.dll	6.0.3790.3959	216 KB	2/17/2007 12:05:40 AM	C:\WINDOWS\system32	Microsoft Corporation
comctl32.dll	5.82.3790.3959	935 KB	2/17/2007 12:09:08 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtrans.dll	6.3.3790.3959	325 KB	2/17/2007 12:18:34 AM	C:\WINDOWS\system32	Microsoft Corporation
dxtrans.dll	6.3.3790.3959	549 KB	2/17/2007 12:18:32 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	16.0.3790.3959	417 KB	2/17/2007 12:29:36 AM	C:\WINDOWS\system32	Microsoft Corporation
iepeers.dll	6.0.3790.3959	362 KB	2/17/2007 12:29:38 AM	C:\WINDOWS\system32	Microsoft Corporation
iesetup.dll	6.0.3790.1830	71 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
ieuinit.inf	Not Available	24 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Not Available
ieexplore.exe	6.0.3790.1830	94 KB	3/25/2005 8:00:00 AM	C:\Program Files\Internet Explorer	Microsoft Corporation
imgutil.dll	6.0.3790.3959	61 KB	2/17/2007 12:30:36 AM	C:\WINDOWS\system32	Microsoft Corporation
inetcp1.cpl	6.0.3790.3959	431 KB	2/17/2007 12:30:40 AM	C:\WINDOWS\system32	Microsoft Corporation
inetcpl.dll	6.0.3790.1830	110 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
inseng.dll	6.0.3790.3959	147 KB	2/17/2007 12:30:52 AM	C:\WINDOWS\system32	Microsoft Corporation

mlang.dll	6.0.3790.3959	686 KB	2/17/2007 12:36:42 AM	C:\WINDOWS\system32	Microsoft Corporation
msencode.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
mshta.exe	6.0.3790.1830	38 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.3959	5,859 KB	2/17/2007 12:38:38 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.tlb	6.0.3790.1830	1,320 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.3959	906 KB	2/17/2007 12:38:42 AM	C:\WINDOWS\system32	Microsoft Corporation
mshtml.dll	6.0.3790.1830	56 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msident.dll	6.0.3790.1830	69 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msidentld.dll	6.0.3790.1830	16 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
msieftp.dll	6.0.3790.3959	369 KB	2/17/2007 12:38:50 AM	C:\WINDOWS\system32	Microsoft Corporation
msrating.dll	6.0.3790.3959	240 KB	2/17/2007 12:39:20 AM	C:\WINDOWS\system32	Microsoft Corporation
mstime.dll	6.0.3790.3959	880 KB	2/17/2007 12:39:26 AM	C:\WINDOWS\system32	Microsoft Corporation
occache.dll	6.0.3790.3959	126 KB	2/17/2007 12:41:48 AM	C:\WINDOWS\system32	Microsoft Corporation
proctexe.ocx	<File Missing>	Not Available	Not Available	Not Available	Not Available
sendmail.dll	6.0.3790.3959	64 KB	2/17/2007 12:54:24 AM	C:\WINDOWS\system32	Microsoft Corporation
shdoclc.dll	6.0.3790.1830	590 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation

shdocvw.dll	6.0.3790.3959	2,381 KB	2/17/2007 12:54:58 AM	C:\WINDOWS\system32	Microsoft Corporation
shfolder.dll	6.0.3790.1830	34 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
shlwapi.dll	6.0.3790.3959	607 KB	2/17/2007 12:55:32 AM	C:\WINDOWS\system32	Microsoft Corporation
tdc.ocx	1.3.0.3130	91 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
url.dll	6.0.3790.1830	40 KB	3/25/2005 8:00:00 AM	C:\WINDOWS\system32	Microsoft Corporation
urlmon.dll	6.0.3790.3959	1,063 KB	2/17/2007 1:00:44 AM	C:\WINDOWS\system32	Microsoft Corporation
webcheck.dll	6.0.3790.3959	439 KB	2/17/2007 1:02:26 AM	C:\WINDOWS\system32	Microsoft Corporation
wininet.dll	6.0.3790.3959	1,163 KB	2/17/2007 1:02:54 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

Changes to the Server

Changes made to the default installation of Windows Server 2003 x64 SP2 Enterprise Edition on the SUT:

OS Updates:
SP2
KB947775 (TCP-IP updates)
KB945778 (NDIS updates)

KB932755 (STORport updates)

OS Services:

The following services were disabled:
Automatic Updates
Computer Browser
Error Reporting Service
Help and Support
Messenger
Print Spooler
Windows Time
Wireless Configuration

Device Drivers:

Drivers were updated/added for the following devices:
Onboard Broadcom Ethernet controllers
Onboard LSI SAS
Onboard ATI video
IBM (Broadcom) Ethernet controllers
IBM (LSI) SAS log controller
IBM (QLogic) FC controllers
IBM Active PCI (hotplug)

OS Settings:

Display = "Adjust for best performance",
1024x768x32
Display hardware acceleration = full

gpedit.msc - Computer Configuration - Windows Settings - Security Settings - Local Policies - User Rights Assignments - policy 'Lock pages in memory' added group 'Administrators'

Update hosts file as needed for benchmark configuration

Microsoft SQL Server Startup Parameters

sqlservr -c -x -T661 -T834 -T836 -T1228 -T3502 -T8011 -T8012 -T8018 -T8019 -T8020 -T8744

where:

-c Start SQL Server independent of the Service Control Manager
-x Disable the keeping of CPU time and cache hit ratio statistic
-T661 - Disable the ghost record removal process
-T834 - Use Large Pages
-T836 - Use "Max Server Memory" setting for the buffer pool
-T1228 - Enable lock partitioning
-T3502 - Display checkpoint information in errorlog
-T8011 - Disable ring buffer for resource monitor

-T8012 - Disable ring buffer for schedulers
-T8018 - Disable exception ring buffer
-T8019 - Disable stack collection for exception ring buffers
-T8020 - Disable working set trimming
-T8744 - Disable pre-fetch

Server Registry Parameters:

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DISK]
"DependOnGroup"=hex(7):53,00,43,00,53,00,49,00,20,00,6d,00,69,00,6e,00,69,00,\
70,00,6f,00,72,00,74,00,00,00,00,00,
"ErrorControl"=dword:00000001
"Group"="SCSI Class"
"Start"=dword:00000000
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="Disk Driver"
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\
```

```
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,64,00,69,00,73,00,6b,00,2e,00,73,\
00,79,00,73,00,00,00
"AutoRunAlwaysDisable"=hex(7):42,00,72,00,6f,00,74,00,68,00,65,00,72,00,20,00,\
```

```
52,00,65,00,6d,00,6f,00,76,00,61,00,62,00,6c,00,65,00,44,00,69,00,73,00,6b,\
00,28,00,55,00,29,00,00,00,00,00
"TimeoutValue"=dword:00000078
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DISK\Enum]
```

```
"0"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&317c9d0a&0&000"
"Count"=dword:0000001a
"NextInstance"=dword:0000001a
"1"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&317c9d0a&0&011"
"2"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&a890e61&0&000"
"3"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&a890e61&0&011"
"4"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&18e37560&0&000"
"5"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&18e37560&0&011"
"6"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0914\5&18e37560&0&020"
"7"="SCSI\Disk&Ven_IBM&Prod_1815_____FAStT&Rev_0
```

914\5&18e37560&0&022"
 "8"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&18e37560&0&024"
 "9"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&18e37560&0&026"
 "10"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&18e37560&0&028"
 "11"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2d8ab0be&0&001"
 "12"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2d8ab0be&0&003"
 "13"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2d8ab0be&0&005"
 "14"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2d8ab0be&0&007"
 "15"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2d8ab0be&0&009"
 "16"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&43f601f&0&001"
 "17"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&43f601f&0&010"
 "18"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2f170704&0&000"
 "19"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&2f170704&0&011"
 "20"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&1cd887c9&0&001"
 "21"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&1cd887c9&0&010"
 "22"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&167ddf5a&0&000"
 "23"="SCSI\Disk&Ven_IBM&Prod_1815_____FAST&Rev_0914\5&167ddf5a&0&011"
 "24"="SCSI\Disk&Ven_IBM-ESXS&Prod_ST973401SS&Rev_B51D\5&22ad4189&0&000f00"
 "25"="SCSI\Disk&Ven_IBM&Prod_ServeRAID-MR10M&Rev_1.20\5&297181f&0&010000"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\I/O System]
 "CountOperations"=dword:00000000

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\LANMANSERVER\Parameters]
 "autodisconnect"=dword:0000000f
 "enableforcedlogoff"=dword:00000001
 "enablesecuritysignature"=dword:00000000
 "requiresecuritysignature"=dword:00000000
 "restrictnullsessaccess"=dword:00000001
 "NullSessionPipes"=hex(7):43,00,4f,00,4d,00,4e,00,41,00,50,00,00,43,00,4f,\

00,4d,00,4e,00,4f,00,44,00,45,00,00,53,00,51,00,4c,00,5c,00,

51,00,55,00,\
 45,00,52,00,59,00,00,00,53,00,50,00,4f,00,4f,00,4c,00,53,00,53,00,00,00,4e,\
 00,45,00,54,00,4c,00,4f,00,47,00,4f,00,4e,00,00,00,4c,00,53,00,41,00,52,00,\
 50,00,43,00,00,00,53,00,41,00,4d,00,52,00,00,00,42,00,52,00,4f,00,57,00,53,\
 00,45,00,52,00,00,00,00,00,
 "NullSessionShares"=hex(7):43,00,4f,00,4d,00,43,00,46,00,47,00,00,00,44,00,4e,\
 00,53,00,24,00,00,00,00,00,
 "ServiceDll"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\
 00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,\

73,00,72,00,76,00,73,00,76,00,63,00,2e,00,64,00,6c,00,6c,00,00,00
 "Lmannounce"=dword:00000000
 "Size"=dword:00000003
 "AdjustedNullSessionPipes"=dword:00000001
 "Guid"=hex:01,ce,8e,56,76,b5,80,4d,98,39,af,c6,ad,e0,7c,e8

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory Management]
 "ClearPageFileAtShutdown"=dword:00000000
 "DisablePagingExecutive"=dword:00000001
 "LargeSystemCache"=dword:00000000
 "NonPagedPoolQuota"=dword:00000000
 "NonPagedPoolSize"=dword:00000000
 "PagedPoolQuota"=dword:00000000
 "PagedPoolSize"=dword:00000000
 "SecondLevelDataCache"=dword:00000000
 "SystemPages"=dword:00000000
 "PagingFiles"=hex(7):43,00,3a,00,5c,00,70,00,61,00,67,00,65,00,66,00,69,00,6c,\

00,65,00,2e,00,73,00,79,00,73,00,20,00,32,00,30,00,34,00,36,00,20,00,34,00,\
 30,00,39,00,32,00,00,00,00,00,
 "PhysicalAddressExtension"=dword:00000001
 "WriteWatch"=dword:00000001
 "DontVerifyRandomDrivers"=dword:00000001

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters]
 "VideoInitTime"=dword:0000057e
 "EnablePrefetcher"=dword:00000002
 "AppLaunchMaxNumPages"=dword:00000fa0
 "AppLaunchMaxNumSections"=dword:000000aa

"AppLaunchTimerPeriod"=hex:80,69,67,ff,ff,ff,ff,ff
 "BootMaxNumPages"=dword:0001f400
 "BootMaxNumSections"=dword:00000ff0
 "BootTimerPeriod"=hex:00,f2,d8,f8,ff,ff,ff,ff
 "MaxNumActiveTraces"=dword:00000008
 "MaxNumSavedTraces"=dword:00000008
 "RootDirPath"="Prefetch"
 "HostingAppList"="DLLHOST.EXE,MMC.EXE,RUNDLL32.EXE"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NDIS]
 "DisplayName"="NDIS System Driver"
 "ErrorControl"=dword:00000001
 "Group"="NDIS Wrapper"
 "Start"=dword:00000000
 "Type"=dword:00000001

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NDIS\MediaTypes]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NDIS\Parameters]
 "ProcessorAffinityMask"=dword:08080808
 "PacketStackSize"=dword:00000004

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\NDIS\Enum]
 "0"="Root\LEGACY_NDIS\0000"
 "Count"=dword:00000001
 "NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QL2300]
 "ErrorControl"=dword:00000001
 "Group"="SCSI miniport"
 "Start"=dword:00000000
 "Tag"=dword:00000028
 "Type"=dword:00000001
 "DisplayName"="QLLogic Fibre Channel SCSI Miniport Driver (wx64 IP)"
 "ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,71,00,6c,00,32,00,33,00,30,00,30,\
 00,2e,00,73,00,79,00,73,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QL2300\Parameters]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QL2300\Parameters\Device]
 "DriverParameter"=""

```

"BusType"=dword:00000006
"MaximumSGList"=dword:000000ff
"NumberOfRequests"=dword:000000ff

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QL2300\Parameters\PnpInterface]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QL2300\Enum]
"0"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&1dd35bbd&0&0000"
"Count"=dword:00000008
"NextInstance"=dword:00000008
"1"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&1dd35bbd&0&0100"
"2"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&15bf98e9&0&0000"
"3"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&15bf98e9&0&0100"
"4"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&167c2c2f&0&0000"
"5"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&167c2c2f&0&0100"
"6"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&2657b4f5&0&0000"
"7"="PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&2657b4f5&0&0100"

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QLDIRECT]
"ErrorControl"=dword:00000001
"Type"=dword:00000001
"Start"=dword:00000002
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\

52,00,49,00,56,00,45,00,52,00,53,00,5c,00,71,00,6c,00,64,00,69,00,72,00,65,\
00,63,00,74,00,2e,00,73,00,79,00,73,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QLDIRECT\Parameters]
"MaxPathsPerDevice"=dword:00000001
"SrbListSize"=dword:00000064
"PerCpuData"=dword:00000001

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QLDIRECT\Security]
"Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,00,02,\

00,1c,00,01,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,0,0,01,00,00,\

```

```

00,00,02,00,88,00,06,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,00,\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,04,00,00,00,00,\

00,14,00,8d,01,02,00,01,01,00,00,00,00,05,06,00,00,00,00,00,14,00,00,01,\

00,00,01,01,00,00,00,00,05,0b,00,00,00,00,18,00,fd,01,02,00,01,02,00,\

00,00,00,00,05,20,00,00,23,02,00,00,01,01,00,00,00,00,05,12,00,00,00,\

01,01,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\QLDIRECT\Enum]
"0"="Root\SCSIADAPTER\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\sqlservr.exe]
"UseLargePages"=dword:00000001

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration]

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node0]
"CPUMask"=dword:00000003

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node1]
"CPUMask"=dword:0000000c

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node10]
"CPUMask"=dword:00300000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node11]
"CPUMask"=dword:00c00000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node2]
"CPUMask"=dword:00000030

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft

```

```

SQL Server\90\NodeConfiguration\Node3]
"CPUMask"=dword:000000c0

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node4]
"CPUMask"=dword:00000300

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node5]
"CPUMask"=dword:00000c00

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node6]
"CPUMask"=dword:00003000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node7]
"CPUMask"=dword:0000c000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node8]
"CPUMask"=dword:00030000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node9]
"CPUMask"=dword:000c0000

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP]
"Enabled"=dword:00000001
"ListenOnAllIPs"=dword:00000001
"NoDelay"=dword:00000000
"KeepAlive"=dword:00007530
"DisplayName"="TCP/IP"

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP\IP1]
"Enabled"=dword:00000001
"Active"=dword:00000001
"TcpPort"="1434[0x1]"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="10.0.0.1"

[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP\IP2]
"Enabled"=dword:00000001
"Active"=dword:00000001
"TcpPort"="1436[0x2]"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="10.0.1.1"

```

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft
SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP\IP3]
"Enabled"=dword:00000001
"Active"=dword:00000001
"TcpPort"="1400"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="127.0.0.1"
```

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft
SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP\IP4]
"Enabled"=dword:00000000
"Active"=dword:00000001
"TcpPort"="1400"
"TcpDynamicPorts"=""
"DisplayName"="Specific IP Address"
"IPAddress"="127.0.0.1"
```

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft
SQL
Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\TCP\IPAll]
"TcpPort"="1400,1415[0x1],1418[0x2],1428[0x4],1435[0x8],14
38[0x10],1448[0x20],1455[0x40],1458[0x80],1468[0x100],1475
[0x200],1478[0x400],1488[0x800]"
"TcpDynamicPorts"=""
"DisplayName"="Any IP Address"
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters]
"NV Hostname"="ibmserv4"
"DataBasePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,
00,52,00,6f,00,6f,\
```

```
00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,
32,00,5c,00,\
```

```
64,00,72,00,69,00,76,00,65,00,72,00,73,00,5c,00,65,00,74,00,63,
00,00,00
```

```
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="ibmserv4"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"ReservedPorts"=hex(7):31,00,34,00,33,00,33,00,2d,00,31,00,34,
,00,33,00,34,00,\
```

```
00,00,33,00,33,00,34,00,33,00,2d,00,33,00,33,00,34,00,33,00,00,
00,00,00
"EnableTCPA"=dword:00000001
"EnableRSS"=dword:00000001
"EnableTCPChimney"=dword:00000001
"MaxUserPort"=dword:00009c40
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters]
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,
00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,00,43,00,38,00,31,00,
35,00,38,00,\
```

```
37,00,38,00,2d,00,38,00,34,00,33,00,46,00,2d,00,34,00,33,00,46,
00,36,00,2d,\
```

```
00,41,00,44,00,33,00,41,00,2d,00,45,00,39,00,43,00,44,00,44,00,
35,00,46,00,\
```

```
37,00,44,00,37,00,42,00,33,00,7d,00,00,00,54,00,63,00,70,00,69,
00,70,00,5c,\
```

```
00,50,00,61,00,72,00,61,00,6d,00,65,00,74,00,65,00,72,00,73,00,
5c,00,49,00,\
```

```
6e,00,74,00,65,00,72,00,66,00,61,00,63,00,65,00,73,00,5c,00,7b,
00,43,00,34,\
```

```
00,33,00,39,00,33,00,45,00,37,00,31,00,2d,00,45,00,43,00,41,00,
34,00,2d,00,\
```

```
34,00,37,00,33,00,31,00,2d,00,42,00,36,00,41,00,32,00,2d,00,41,
00,46,00,34,\
```

```
00,35,00,44,00,45,00,35,00,31,00,46,00,38,00,36,00,44,00,7d,00,
00,00,00,00
```

```
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:78,58,81,7c,3f,84,f6,43,ad,3a,e9,cd,d5,f7,d7,
b3,71,3e,39,c4,\
a4,ec,31,47,b6,a2,af,45,de,51,f8,6d
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{07CE071E-8F92-40D9-8B92-
C298E4E9E51B}]
```

```
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,
00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,00,37,00,43,00,45,00,
30,00,37,00,\
```

```
31,00,45,00,2d,00,38,00,46,00,39,00,32,00,2d,00,34,00,30,00,44,
00,39,00,2d,\
```

```
00,38,00,42,00,39,00,32,00,2d,00,43,00,32,00,39,00,38,00,45,00,
34,00,45,00,\
39,00,45,00,35,00,31,00,42,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{123B8DFD-7077-4FD4-8E50-
CD60966A90F1}]
"LLInterface"=""
```

```
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,
00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,32,00,33,00,42,00,
38,00,44,00,\
```

```
46,00,44,00,2d,00,37,00,30,00,37,00,37,00,2d,00,34,00,46,00,44,
00,34,00,2d,\
```

```
00,38,00,45,00,35,00,30,00,2d,00,43,00,44,00,36,00,30,00,39,00,
36,00,36,00,\
```

```
41,00,39,00,30,00,46,00,31,00,7d,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{1967528B-E313-4815-AAF1-
344282EB2CCC}]
"LLInterface"=""
```

```
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,6
1,00,72,00,61,00,\
```

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,
00,72,00,66,\
```

```
00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,00,39,00,36,00,37,00,
35,00,32,00,\
```

```
38,00,42,00,2d,00,45,00,33,00,31,00,33,00,2d,00,34,00,38,00,31,
00,35,00,2d,\
```

```
00,41,00,41,00,46,00,31,00,2d,00,33,00,34,00,34,00,32,00,38,00,
```


32,00,45,00,\
42,00,32,00,43,00,43,00,43,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{2E4AFC59-853E-4456-9217-B8C70DAA489E}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,00,45,00,34,00,41,00,46,00,43,00,\

35,00,39,00,2d,00,38,00,35,00,33,00,45,00,2d,00,34,00,34,00,35,00,36,00,2d,\

00,39,00,32,00,31,00,37,00,2d,00,42,00,38,00,43,00,37,00,30,00,44,00,41,00,\

41,00,34,00,38,00,39,00,45,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{3CBDE2A0-D65E-40B0-A303-BCB91B615FD4}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,33,00,43,00,42,00,44,00,45,00,32,00,\

41,00,30,00,2d,00,44,00,36,00,35,00,45,00,2d,00,34,00,30,00,42,00,30,00,2d,\

00,41,00,33,00,30,00,33,00,2d,00,42,00,43,00,42,00,39,00,31,00,42,00,36,00,\

31,00,35,00,46,00,44,00,34,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{43AC19E5-66D3-4F76-BD85-CBB39D8D6B42}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,34,00,33,00,41,00,43,00,

31,00,39,00,\

45,00,35,00,2d,00,36,00,36,00,44,00,33,00,2d,00,34,00,46,00,37,00,36,00,2d,\

00,42,00,44,00,38,00,35,00,2d,00,43,00,42,00,42,00,33,00,39,00,44,00,38,00,\

44,00,36,00,42,00,34,00,32,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{5BE2D40D-C4C1-4EC4-91D6-3142CEBEEF93}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,35,00,42,00,45,00,32,00,44,00,34,00,\

30,00,44,00,2d,00,43,00,34,00,43,00,31,00,2d,00,34,00,45,00,43,00,34,00,2d,\

00,39,00,31,00,44,00,36,00,2d,00,33,00,31,00,34,00,32,00,43,00,45,00,42,00,\

45,00,45,00,46,00,39,00,33,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{6A87BB84-850D-44B9-A7BB-F4A2245CFC37}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,00,41,00,38,00,37,00,42,00,42,00,\

38,00,34,00,2d,00,38,00,35,00,30,00,44,00,2d,00,34,00,34,00,42,00,39,00,2d,\

00,41,00,37,00,42,00,42,00,2d,00,46,00,34,00,41,00,32,00,32,00,34,00,35,00,\

43,00,46,00,43,00,33,00,37,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{7E94B524-595F-4777-8D28-727D9F5DF1A1}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,34,00,33,00,41,00,43,00,

1,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,00,45,00,39,00,34,00,42,00,35,00,\

32,00,34,00,2d,00,35,00,39,00,35,00,46,00,2d,00,34,00,37,00,37,00,37,00,2d,\

00,38,00,44,00,32,00,38,00,2d,00,37,00,32,00,37,00,44,00,39,00,46,00,35,00,\

44,00,46,00,31,00,41,00,31,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{8E5CE6CA-1C43-42A2-99DC-E5296E40F7F1}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,38,00,45,00,35,00,43,00,45,00,36,00,\

43,00,41,00,2d,00,31,00,43,00,34,00,33,00,2d,00,34,00,32,00,41,00,32,00,2d,\

00,39,00,39,00,44,00,43,00,2d,00,45,00,35,00,32,00,39,00,36,00,45,00,34,00,\

30,00,46,00,37,00,46,00,31,00,7d,00,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{909C9159-3CEE-49E6-A87D-C79A7C31D1B8}]\n\"LLInterface\"=\"\"\n\"IpConfig\"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,39,00,30,00,39,00,43,00,39,00,31,00,\

35,00,39,00,2d,00,33,00,43,00,45,00,45,00,2d,00,34,00,39,00,45,00,36,00,2d,\

00,41,00,38,00,37,00,44,00,2d,00,43,00,37,00,39,00,41,00,37,00,43,00,33,00,\

31,00,44,00,31,00,42,00,38,00,7d,00,00,00,00

<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{AB0D6FDC-168F-41C6-8ECA-E4B5C9C71858}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	<pre>31,00,45,00,2d,00,34,00,46,00,36,00,31,00,2d,00,34,00,35,00,39,00,30,00,2d,\ 00,42,00,46,00,30,00,38,00,2d,00,31,00,44,00,41,00,35,00,39,00,37,00,32,00,\ 34,00,33,00,39,00,30,00,37,00,7d,00,00,00,00,00</pre>	<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\ 00,61,00,63,00,65,00,73,00,5c,00,7b,00,45,00,38,00,42,00,45,00,39,00,30,00,\</pre>
<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\ 00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,00,42,00,30,00,44,00,36,00,46,00,\</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{C373E2A4-9D66-4192-A377-F55C0D1EF631}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	<pre>43,00,41,00,2d,00,41,00,34,00,34,00,33,00,2d,00,34,00,36,00,43,00,44,00,2d,\ 00,39,00,43,00,37,00,32,00,2d,00,43,00,36,00,46,00,30,00,45,00,46,00,33,00,\ 42,00,33,00,39,00,44,00,38,00,7d,00,00,00,00,00</pre>
<pre>44,00,43,00,2d,00,31,00,36,00,38,00,46,00,2d,00,34,00,31,00,43,00,36,00,2d,\</pre>	<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\DNSRegisteredAdapters]</pre>
<pre>00,38,00,45,00,43,00,41,00,2d,00,45,00,34,00,42,00,35,00,43,00,39,00,43,00,\ 37,00,31,00,38,00,35,00,38,00,7d,00,00,00,00,00</pre>	<pre>00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,33,00,37,00,33,00,45,00,32,00,\</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces]</pre>
<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{B7B0F592-B810-49D6-AEC2-0C789B3B0F60}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	<pre>41,00,34,00,2d,00,39,00,44,00,36,00,36,00,2d,00,34,00,31,00,39,00,32,00,2d,\ 00,41,00,33,00,37,00,37,00,2d,00,46,00,35,00,35,00,43,00,30,00,44,00,31,00,\ 45,00,46,00,36,00,33,00,31,00,7d,00,00,00,00,00</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{07CE071E-8F92-40D9-8B92-C298E4E9E51B}] "UseZeroBroadcast"=dword:00000000 "EnableDeadGWDetect"=dword:00000001 "EnableDHCP"=dword:00000001 "IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00,00 "SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,0,00,00,00,00 "DefaultGateway"=hex(7):00,00 "DefaultGatewayMetric"=hex(7):00,00 "NameServer"="" "Domain"="" "RegistrationEnabled"=dword:00000001 "RegisterAdapterName"=dword:00000000 "TCPAllowedPorts"=hex(7):30,00,00,00,00,00 "UDPAllowedPorts"=hex(7):30,00,00,00,00,00 "RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00 "NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,30,00,\ 34,00,00,00,00,00</pre>
<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\ 00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,00,37,00,42,00,30,00,46,00,35,00,\</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{C46575DB-972C-452A-B499-501524D93CC9}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	
<pre>39,00,32,00,2d,00,42,00,38,00,31,00,30,00,2d,00,34,00,39,00,44,00,36,00,2d,\</pre>	<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\</pre>	
<pre>00,41,00,45,00,43,00,32,00,2d,00,30,00,43,00,37,00,38,00,39,00,42,00,33,00,\ 42,00,30,00,46,00,36,00,30,00,7d,00,00,00,00,00</pre>	<pre>00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,00,34,00,36,00,35,00,37,00,35,00,\</pre>	
<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{BA0E9D1E-4F61-4590-BF08-1DA597243907}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	<pre>44,00,42,00,2d,00,39,00,37,00,32,00,43,00,2d,00,34,00,35,00,32,00,41,00,2d,\ 00,42,00,34,00,39,00,39,00,2d,00,35,00,30,00,31,00,35,00,32,00,34,00,44,00,\ 39,00,33,00,43,00,43,00,39,00,7d,00,00,00,00,00</pre>	
<pre>6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,49,00,6e,00,74,00,65,00,72,00,66,\</pre>	<pre>[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Adapters\{E8BE90CA-A443-46CD-9C72-C6F0EF3B39D8}] "LLInterface"="" "IpConfig"=hex(7):54,00,63,00,70,00,69,00,70,00,5c,00,50,00,61,00,72,00,61,00,\</pre>	
<pre>00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,00,41,00,30,00,45,00,39,00,44,00,\</pre>		

```
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{1967528B-E313-4815-AAF1-344282EB2CCC}]
```

```
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,32,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,\
32,00,00,00,00,00
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:4676fe62
"T1"=dword:4677056a
"T2"=dword:46770ab0
"LeaseTerminatesTime"=dword:46770c72
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{2E4AFC59-853E-4456-9217-B8C70DAA489E}]
```

```
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,30,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
```

```
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,\
32,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{3CBDE2A0-D65E-40B0-A303-BCB91B615FD4}]
```

```
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,30,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,\
35,00,00,00,00,00
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:48728037
"T1"=dword:4872873f
"T2"=dword:48728c85
"LeaseTerminatesTime"=dword:48728e47
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{43AC19E5-66D3-4F76-BD85-CBB39D8D6B42}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
```

```
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,31,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,30,00,30,00,\
33,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{5BE2D40D-C4C1-4EC4-91D6-3142CEBEEF93}]
```

```
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,30,00,2e,00,30,00,2e,00,31,00,00,00,\
00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"DhcpClassIdBin"=hex:
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{6A87BB84-850D-44B9-A7BB-F4A2245CF37}]
```

```
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,0,0,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
```



```
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{C373E2A4-9D66-4192-A377-F55C0D1EF631}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
"DhcpClassIdBin"=hex:
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:4676fd9e
"T1"=dword:467704a6
"T2"=dword:467709ec
"LeaseTerminatesTime"=dword:46770bae
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{C4393E71-ECA4-4731-B6A2-AF45DE51F86D}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
```

```
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{C46575DB-972C-452A-B499-501524D93CC9}]
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,30,00,2e,00,31,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00,30,00,30,00,30,00,34,00,00,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Interfaces\{E8BE90CA-A443-46CD-9C72-C6F0EF3B39D8}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,00,2e,00,31,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,00,32,00,35,00,35,00,2e,00,32,00,35,\
00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"RegistrationEnabled"=dword:00000001
"RegisterAdapterName"=dword:00000000
"TCPAllowedPorts"=hex(7):30,00,00,00,00,00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00,00,00,00
"NTEContextList"=hex(7):00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\PersistentRoutes]
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,\
6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,\
00,77,00,73,00,68,00,74,00,63,00,70,00,69,00,70,00,2e,00,64,00,6c,00,6c,00,\
00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,02,\
00,00,00,01,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00,\
00,00,00,00,00,06,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00,00,\
00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,00,00,02,00,00,\
00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00
```

Storage Configuration Parameters

ServeRAID-MR10M Controller Configuration

Adapter #0

```
=====
Versions
=====
```

Product Name : IBM ServeRAID-MR10M SAS/SATA
Controller
Serial No : L000983407

FW Package Build: 9.0.1-0030

Mfg. Data

Mfg. Date : 08/25/07
Rework Date : 00/00/00
Revision No :

Image Versions In Flash:

FW Version : 1.20.12-0475
BIOS Version : 2.01.00
WebBIOS Version : 1.1-47-e_15-Rel
Ctrl-R Version : 1.02-012B
Boot Block Version : 1.00.00.01-0011

Pending Images In Flash

None

PCI Info

Vendor Id : 1000
Device Id : 0060
SubVendorId : 1014
SubDeviceId : 0379

Host Interface : PCIE

Number of Frontend Port: 0
Device Interface : PCIE

Number of Backend Port: 8

Port : Address
0 500a0b81e1b0b000
1 0000000000000000
2 0000000000000000
3 0000000000000000
4 500a0b8245b09000
5 0000000000000000
6 0000000000000000
7 0000000000000000

HW Configuration

SAS Address : 500605b00036d1b0
BBU : Present
Alarm : Present
NVRAM : Present
Serial Debugger : Present
Memory : Present
Flash : Present

Settings

Current Time : 15:28:43 8/28, 2008
Predictive Fail Poll Interval : 65535sec
Interrupt Throttle Active Count : 16
Interrupt Throttle Completion : 50us
Rebuild Rate : 100%
PR Rate : 0%
Resynch Rate : 100%
Check Consistency Rate : 0%
Reconstruction Rate : 100%
Cache flush interval : 255s
Max drives to spinup at one time : 2
Delay among spinup groups : 12s
Physical drive coercion mode : 1GB
Cluster mode : Disabled
Alarm : Disabled
Auto Rebuild : Enabled
Battery Warning : Enabled

Ecc Bucket Size : 0
Ecc Bucket Leak Rate : 0 Minutes
Restore HotSpare On Insertion : Disabled
Expose Enclosure Devices : Disabled
Maintain PD Fail History : Disabled
Host Request Reordering : Enabled

Capabilities

RAID Level Supported : RAID0, RAID1, RAID5,
RAID6, RAID10, RAID50, RAID60
Supported Drives : SAS, SATA

Allowed Mixing:
Mix In Enclosure Allowed, Mix In VD Allowed

Status

ECC Bucket Count : 0

Limitations

Max Arms Per VD : 32
Max Spans Per VD : 8
Max Arrays : 128
Max Number of VDs : 64
Max Parallel Commands : 1008
Max SGE Count : 80
Max Data Transfer Size : 8192 sectors
Max Strips PerIO : 42
Min Stripe Size : 8kB
Max Stripe Size : 1024kB

Device Present

Virtual Drives : 1
Degraded : 0
Offline : 0
Physical Devices : 18
Disks : 16
Critical Disks : 0
Failed Disks : 0

Supported Adapter Operations

Rebuild Rate : Yes
CC Rate : Yes
BGI Rate : Yes
Reconstruct Rate : Yes
Patrol Read Rate : Yes
Alarm Control : Yes
Cluster Support : No
BBU : Yes
Spanning : Yes
Dedicated Hot Spare : Yes
Revertible Hot Spares : Yes
Foreign Config Import : Yes
Self Diagnostic : Yes
Allow Mixed Redundancy On Array : No
Global Hot Spares : Yes
Deny SCSI Passthrough : No
Deny SMP Passthrough : No
Deny STP Passthrough : No

Supported VD Operations

Read Policy : Yes
Write Policy : Yes
IO Policy : Yes
Access Policy : Yes
Disk Cache Policy : Yes
Reconstruction : Yes
Deny Locate : No
Deny CC : No

Supported PD Operations

Force Online : Yes
Force Offline : Yes
Force Rebuild : Yes
Deny Force Failed : No
Deny Force Good/Bad : No
Deny Missing Replace : No
Deny Clear : No
Deny Locate : No

Error Counters

Memory Correctable Errors : 0

Memory Uncorrectable Errors : 0

Cluster Information

Cluster Permitted : No
Cluster Active : No

Default Settings

Phy Polarity : 0
Phy PolaritySplit : 0
Background Rate : 30
Stripe Size : 128kB
Flush Time : 4 seconds
Write Policy : WB
Read Policy : None
Cache When BBU Bad : Disabled
Cached IO : No
SMART Mode : Mode 6
Alarm Disable : Yes
Coercion Mode : 1GB
ZCR Config : Unkown
Dirty LED Shows Drive Activity : No
BIOS Continue On Error : No
Spin Down Mode : None
Allowed Device Type : SAS/SATA Mix
Allow Mix In Enclosure : Yes
Allow Mix In VD : Yes
Allow SATA In Cluster : No
Max Chained Enclosures : 16
Disable Ctrl-R : Yes
Enable Web BIOS : Yes
Direct PD Mapping : No
BIOS Enumerate VDs : Yes
Restore Hot Spare On Insertion : No
Expose Enclosure Devices : Yes
Maintain PD Fail History : Yes
Disable Puncturing : Yes
Zero Based Enclosure Enumeration: No
PreBoot CLI Enabled : Yes
LED Show Drive Activity : No
Cluster Disable : Yes
SAS Disable : No

Number of enclosures on adapter 0 -- 2

Enclosure 0:

Device ID : 8
Number of Slots : 12
Number of Power Supplies : 2
Number of Fans : 4
Number of Temperature Sensors : 4
Number of Alarms : 0

Number of SIM Modules : 2
Number of Physical Drives : 8
Status : Normal
Position : 1
Connector Name : Int.Ports 0-3

Enclosure 1:

Device ID : 9
Number of Slots : 12
Number of Power Supplies : 2
Number of Fans : 4
Number of Temperature Sensors : 4
Number of Alarms : 0
Number of SIM Modules : 2
Number of Physical Drives : 8
Status : Normal
Position : 1
Connector Name : Int.Ports 4-7

Adapter 0 -- Virtual Drive Information:

Virtual Disk: 0 (target id: 0)
Name:LOG
RAID Level: Primary-1, Secondary-0, RAID Level Qualifier-0
Size:3807064MB
State: Optimal
Stripe Size: 64kB
Number Of Drives:2
Span Depth:8
Default Cache Policy: WriteThrough, ReadAdaptive, Direct,
Write Cache OK if Bad BBU
Current Cache Policy: WriteThrough, ReadAdaptive, Direct,
Write Cache OK if Bad BBU
Access Policy: Read/Write
Disk Cache Policy: Enabled

DS4800 Controller Configuration

PROFILE FOR STORAGE SUBSYSTEM: C4_DS4800_73GB
(8/28/08 3:21:12 PM)

SUMMARY-----

Number of controllers: 2
High performance tier controllers: Enabled
Number of logical drive groups: 2

Total number of logical drives used: 3
Number of standard logical drives: 2
Number of access logical drives: 1
Total number of logical drives allowed: 2048

FlashCopy Logical Drives: Enabled
Number of flashcopies used: 0
Number of flashcopies allowed: 2
Number of flashcopies allowed per base logical drive: 2

Remote Mirroring: Disabled/Deactivated
Number of mirrors used: 0
Number of mirrors allowed: 0

VolumeCopy: Disabled
Number of copies used: 0
Number of copies allowed: 2048

Number of drives: 192
Mixed drive types: Disabled
Current drive type(s): Fibre (192)
Total hot spare drives: 0
Standby: 0
In use: 0

Number of drive enclosures: 12
Number of drive enclosures allowed: 16

Storage Partitioning: Enabled
Number of partitions used: 0
Number of partitions allowed: 8
Number of logical drives allowed per partition: 256

Access logical drive: None mapped
Default host OS: Windows 2000/Server 2003 Non-Clustered (Host OS index 2)

Current configuration
Firmware version: 07.10.23.00
NVS RAM version: N1815D48R1010V05
EMW version: 10.10.G5.04
AMW version: 10.10.G5.04
Pending configuration

Staged firmware download supported: Yes
Firmware version: None
NVS RAM version: None
Transferred on: None
Controller enclosure audible alarm: Disabled

NVS RAM configured for batteries: Yes

Start cache flushing at (in percentage): 1
Stop cache flushing at (in percentage): 1
Cache block size (in KB): 16

Media scan frequency (in days): Disabled

Failover alert delay (in minutes): 5

Feature enable identifier:
353033313120353038303620482D74EC

Feature pack: Generic
Feature pack submodel ID: 2

Storage Subsystem world-wide identifier (ID):
600A0B800047154E00000000482D7541

CONTROLLERS-----
Number of controllers: 2

Controller in Enclosure 85, Slot A

Status: Online

Current configuration

Firmware version: 07.10.23.00
Appware version: 07.10.23.00
Bootware version: 07.10.23.00
NVS RAM version: N1815D48R1010V05

Pending configuration

Firmware version: None
Appware version: None
Bootware version: None
NVS RAM version: None
Transferred on: None

Current ID (ALPA)

On drive channel 1: 125/0x1
Replacement part number: 44X2427
Board ID: 6091
Submodel ID: 2
Product ID: 1815 FAST
Product revision: 0914
Serial number: SP74850311
Date of manufacture: December 7, 2007

Cache
Total data cache: 4096 MB
Accessible data cache: 4096 MB
Processor cache: 508 MB
Host Interface Board
Board ID: 0901
Host card status: Optimal
Serial number: SN SP74442393
Date of manufacture: November 1, 2007
Part number: PN 12380-05
Vendor: VN IBM
Date/Time: Thu Aug 28 15:20:39 EDT 2008

Associated Logical Drives (* = Preferred Owner):
LD_1*

Controller host name: target
Remote login: Disabled

Ethernet port: 1
MAC address: 00:a0:b8:47:15:4e
Network configuration: Static
IP address: 192.168.40.208
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0

Ethernet port: 2
MAC address: 00:a0:b8:47:15:4f
Network configuration: Static
IP address: 192.168.129.101
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0

Drive interface: Fibre
Channel: 1
Port: 4, 3, Out, In, In, Out, In, In, Out
Current ID: 125/0x1
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Drive interface: Fibre
Channel: 2
Port: 2, 1, Out, In, Out, In, Out, In, Out
Current ID: 125/0x1
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Drive interface: Fibre
Channel: 3
Port: 5, 6, In, Out, In, Out, In, Out
Current ID: 125/0x1
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Drive interface: Fibre
Channel: 4
Port: 7, 8, In, Out, Out, In, Out, In, Out, In, Out
Current ID: 125/0x1
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Host interface: Fibre
Channel: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 0/0xEF
NL-Port ID: 0x010500
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up
Topology: Fabric Attach
World-wide port identifier: 20:14:00:a0:b8:47:15:4e
World-wide node identifier: 20:04:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre
Channel: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 1/0xE8
NL-Port ID: 0xFFFFFFFF
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Down
Topology: Not Available
World-wide port identifier: 20:24:00:a0:b8:47:15:4e
World-wide node identifier: 20:04:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre
Channel: 3
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 2/0xE4
NL-Port ID: 0xFFFFFFFF
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps

Data rate control: Auto
Link status: Down
Topology: Not Available
World-wide port identifier: 20:34:00:a0:b8:47:15:4e
World-wide node identifier: 20:04:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre
Channel: 4
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 3/0xE2
NL-Port ID: 0xFFFFFFFF
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Down
Topology: Not Available
World-wide port identifier: 20:44:00:a0:b8:47:15:4e
World-wide node identifier: 20:04:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Controller in Enclosure 85, Slot B

Status: Online

Current configuration

Firmware version: 07.10.23.00
Appware version: 07.10.23.00
Bootware version: 07.10.23.00
NVS RAM version: N1815D48R1010V05

Pending configuration

Firmware version: None
Appware version: None
Bootware version: None
NVS RAM version: None
Transferred on: None

Current ID (ALPA)

On drive channel 1: 124/0x2
Replacement part number: 44X2427
Board ID: 6091
Submodel ID: 2
Product ID: 1815 FAST
Product revision: 0914
Serial number: SP74850806
Date of manufacture: December 5, 2007

Cache

Total data cache: 4096 MB
Accessible data cache: 4096 MB
Processor cache: 508 MB

Host Interface Board

Board ID: 0901
Host card status: Optimal
Serial number: SN SP74746380

Date of manufacture: November 1, 2007
Part number: PN 12380-05
Vendor: VN IBM
Date/Time: Thu Aug 28 15:20:40 EDT 2008

Associated Logical Drives (* = Preferred Owner): LD_2*

Controller host name: target
Remote login: Disabled

Ethernet port: 1
MAC address: 00:a0:b8:47:18:f2
Network configuration: Static
IP address: 192.168.40.209
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0

Ethernet port: 2
MAC address: 00:a0:b8:47:18:f3
Network configuration: Static
IP address: 192.168.129.102
Subnet mask: 255.255.255.0
Gateway: 0.0.0.0

Drive interface: Fibre
Channel: 1
Port: 4, 3, Out, In, In, Out, In, In, Out
Current ID: 124/0x2
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Drive interface: Fibre
Channel: 2
Port: 2, 1, Out, In, Out, In, Out, In, Out
Current ID: 124/0x2
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Drive interface: Fibre
Channel: 3
Port: 5, 6, In, Out, In, Out, In, Out
Current ID: 124/0x2
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto

Link status: Up

Drive interface: Fibre
Channel: 4
Port: 7, 8, In, Out, Out, In, Out, In, Out, In, Out
Current ID: 124/0x2
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up

Host interface: Fibre
Channel: 1
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 4/0xE1
NL-Port ID: 0x010600
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Up
Topology: Fabric Attach
World-wide port identifier: 20:15:00:a0:b8:47:15:4e
World-wide node identifier: 20:05:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre
Channel: 2
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 5/0xE0
NL-Port ID: 0xFFFFFFFF
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Down
Topology: Not Available
World-wide port identifier: 20:25:00:a0:b8:47:15:4e
World-wide node identifier: 20:05:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre
Channel: 3
Current ID: Not applicable/0xFFFFFFFF
Preferred ID: 6/0xDC
NL-Port ID: 0xFFFFFFFF
Maximum data rate: 4 Gbps
Current data rate: 4 Gbps
Data rate control: Auto
Link status: Down
Topology: Not Available
World-wide port identifier: 20:35:00:a0:b8:47:15:4e
World-wide node identifier: 20:05:00:a0:b8:47:15:4e
Part type: HPFC-5700 revision 5

Host interface: Fibre

Channel: 4
 Current ID: Not applicable/0xFFFFFFFF
 Preferred ID: 7/0xDA
 NL-Port ID: 0xFFFFFFFF
 Maximum data rate: 4 Gbps
 Current data rate: 4 Gbps
 Data rate control: Auto
 Link status: Down
 Topology: Not Available
 World-wide port identifier: 20:45:00:a0:b8:47:15:4e
 World-wide node identifier: 20:05:00:a0:b8:47:15:4e
 Part type: HPFC-5700 revision 5

ARRAYS-----

Number of logical drive groups: 2

Name: Array_1

Status: Optimal

Capacity 6.362 TB

RAID level: 0

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot A

Associated logical drives and free capacity

Logical Drive Capacity

LD_1 6.362 TB

Free Capacity 80.250 MB

Associated drives - present (in piece order)

Enclosure Slot

0 1
 0 2
 0 3
 0 4
 0 5
 0 6
 0 7
 0 8
 0 9
 0 10

0 11
 0 12
 0 13
 0 14
 0 15
 0 16
 1 1
 1 2
 1 3
 1 4
 1 5
 1 6
 1 7
 1 8
 3 1
 3 2
 3 3
 3 4
 3 5
 3 6
 3 7
 3 8
 3 9
 3 10
 3 11
 3 12
 3 13
 3 14
 3 15
 3 16
 4 1
 4 2
 4 3
 4 4
 4 5
 4 6
 4 7
 4 8
 4 9
 4 10
 4 11
 4 12
 4 13
 4 14
 4 15
 4 16

7 1
 7 2
 7 3
 7 4
 7 5
 7 6
 7 7
 7 8
 9 1
 9 2
 9 3
 9 4
 9 5
 9 6
 9 7
 9 8
 9 9
 9 10
 9 11
 9 12
 9 13
 9 14
 9 15
 9 16
 10 1
 10 2
 10 3
 10 4
 10 5
 10 6
 10 7
 10 8

Name: Array_2

Status: Optimal

Capacity 6.362 TB

RAID level: 0

Drive type: Fibre Channel

Enclosure loss protection: No

Current owner: Controller in slot B

Associated logical drives and free capacity

Logical Drive Capacity

LD_2 6.362 TB

Free Capacity 80.250 MB

Associated drives - present (in piece order)

Enclosure Slot

2	1
2	2
2	3
2	4
2	5
2	6
2	7
2	8
2	9
2	10
2	11
2	12
2	13
2	14
2	15
2	16
1	9
1	10
1	11
1	12
1	13
1	14
1	15
1	16
5	1
5	2
5	3
5	4
5	5
5	6
5	7
5	8
5	9
5	10
5	11
5	12
5	13
5	14
5	15
5	16
4	9
4	10
4	11
4	12
4	13
4	14

4	15
4	16
8	1
8	2
8	3
8	4
8	5
8	6
8	7
8	8
8	9
8	10
8	11
8	12
8	13
8	14
8	15
8	16
7	9
7	10
7	11
7	12
7	13
7	14
7	15
7	16
11	1
11	2
11	3
11	4
11	5
11	6
11	7
11	8
11	9
11	10
11	11
11	12
11	13
11	14
11	15
11	16
10	9
10	10
10	11
10	12
10	13
10	14
10	15
10	16

STANDARD LOGICAL DRIVES-----

SUMMARY

Number of standard logical drives: 2

See other Logical Drives sub-tabs for premium feature information.

NAME	STATUS	CAPACITY	RAID LEVEL	ARRAY
LD_1	Optimal	6.362 TB	0	Array_1 Fibre
LD_2	Optimal	6.362 TB	0	Array_2 Fibre

DETAILS

Logical Drive name: LD_1

Logical Drive status: Optimal

Capacity: 6.362 TB
 Logical Drive world-wide identifier:
 60:0a:0b:80:00:47:15:4e:00:00:0d:ef:48:ad:67:92
 Subsystem ID (SSID): 0
 Associated array: Array_1
 RAID level: 0

Drive type: Fibre Channel
Enclosure loss protection: No

Preferred owner: Controller in slot A
Current owner: Controller in slot A

Segment size: 128 KB
Capacity reserved for future segment size changes: Yes
Maximum future segment size: 2,048 KB
Modification priority: High

Read cache: Enabled
Write cache: Enabled
Write cache without batteries: Enabled
Write cache with mirroring: Disabled
Flush write cache after (in seconds): 10.00
Dynamic cache read prefetch: Enabled

Enable background media scan: Disabled
Media scan with redundancy check: Disabled

Logical Drive name: LD_2

Logical Drive status: Optimal

Capacity: 6.362 TB
 Logical Drive world-wide identifier: 60:0a:0b:80:00:47:18:f2:00:00:0d:b8:48:ad:67:de
 Subsystem ID (SSID): 1
 Associated array: Array_2
 RAID level: 0

Drive type: Fibre Channel
 Enclosure loss protection: No

Preferred owner: Controller in slot B
 Current owner: Controller in slot B

Segment size: 128 KB
 Capacity reserved for future segment size changes: Yes
 Maximum future segment size: 2,048 KB
 Modification priority: High

Read cache: Enabled
 Write cache: Enabled
 Write cache without batteries: Enabled
 Write cache with mirroring: Disabled
 Flush write cache after (in seconds): 10.00
 Dynamic cache read prefetch: Enabled

Enable background media scan: Disabled
 Media scan with redundancy check: Disabled

SNAPSHOT REPOSITORY LOGICAL DRIVES-----

Number of flashcopy repositories: 0

SNAPSHOT LOGICAL DRIVES-----

Number of flashcopy logical drives: 0

MISSING LOGICAL DRIVES-----

Number of missing logical drives: 0

DRIVE CHANNELS-----

SUMMARY

CHANNEL PORT	STATUS	CTRL A
LINK CTRL B LINK		

1	4,3,Out,In,In,Out,In,In,Out	Optimal Up
Failed		
2	2,1,Out,In,Out,In,Out,In,Out	Optimal Up
Failed		
3	5,6,In,Out,In,Out,In,Out	Optimal Up
Failed		
4	7,8,In,Out,Out,In,Out,In,Out,In,Out	Optimal Up
Failed		

DETAILS

DRIVE CHANNEL 1

Port: 4, 3, Out, In, In, Out, In, In, Out
 Status: Optimal
 Max. Rate: 4 Gbps
 Current Rate: 4 Gbps
 Rate Control: Auto
 Controller A link status: Up
 Controller B link status: Failed

DRIVE COUNTS

Total # of attached drives: 96
 Connected to: A, Port 4
 Attached drives: 48
 Drive enclosure: 1 (16 drives)
 Drive enclosure: 2 (16 drives)
 Drive enclosure: 0 (16 drives)
 Connected to: A, Port 3
 Attached drives: 48
 Drive enclosure: 3 (16 drives)
 Drive enclosure: 5 (16 drives)
 Drive enclosure: 4 (16 drives)

CUMULATIVE ERROR COUNTS

Controller A

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 1525
 Drive detected errors: 21001
 Timeout errors: 4
 Link down errors: N/A
 Total I/O count: 532475267

Controller B

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 0
 Drive detected errors: 0

Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 0

DRIVE CHANNEL 2

Port: 2, 1, Out, In, Out, In, Out, In, Out
 Status: Optimal
 Max. Rate: 4 Gbps
 Current Rate: 4 Gbps
 Rate Control: Auto
 Controller A link status: Up
 Controller B link status: Failed

DRIVE COUNTS

Total # of attached drives: 96
 Connected to: A, Port 2
 Attached drives: 48
 Drive enclosure: 7 (16 drives)
 Drive enclosure: 6 (16 drives)
 Drive enclosure: 8 (16 drives)
 Connected to: A, Port 1
 Attached drives: 48
 Drive enclosure: 10 (16 drives)
 Drive enclosure: 11 (16 drives)
 Drive enclosure: 9 (16 drives)

CUMULATIVE ERROR COUNTS

Controller A

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 39
 Drive detected errors: 1498
 Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 538395114

Controller B

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 0

DRIVE CHANNEL 3

Port: 5, 6, In, Out, In, Out, In, Out

Status: Optimal
 Max. Rate: 4 Gbps
 Current Rate: 4 Gbps
 Rate Control: Auto
 Controller A link status: Up
 Controller B link status: Failed

DRIVE COUNTS

Total # of attached drives: 0

CUMULATIVE ERROR COUNTS

Controller A

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 1399
 Drive detected errors: 20578
 Timeout errors: 17
 Link down errors: N/A
 Total I/O count: 531986356

Controller B

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 0

DRIVE CHANNEL 4

Port: 7, 8, In, Out, Out, In, Out, In, Out, In, Out
 Status: Optimal
 Max. Rate: 4 Gbps
 Current Rate: 4 Gbps
 Rate Control: Auto
 Controller A link status: Up
 Controller B link status: Failed

DRIVE COUNTS

Total # of attached drives: 96
 Connected to: B, Port 7
 Attached drives: 48
 Drive enclosure: 7 (16 drives)
 Drive enclosure: 6 (16 drives)
 Drive enclosure: 8 (16 drives)
 Connected to: B, Port 8
 Attached drives: 48

Drive enclosure: 10 (16 drives)
 Drive enclosure: 11 (16 drives)
 Drive enclosure: 9 (16 drives)

CUMULATIVE ERROR COUNTS

Controller A

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 2
 Drive detected errors: 1410
 Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 535380111

Controller B

Baseline time set: 8/28/08 3:21:26 PM
 Sample period (hh:mm:ss): 00:00:00
 Controller detected errors: 0
 Drive detected errors: 0
 Timeout errors: 0
 Link down errors: N/A
 Total I/O count: 0

MAPPINGS (Storage Partitioning - Enabled (0 of 8 used))-----

Logical Drive Name	LUN	Controller	Accessible by
Logical Drive status			
LD_1	0	A	Storage Subsystem Optimal
LD_2	1	B	Storage Subsystem Optimal

TOPOLOGY DEFINITIONS

STORAGE SUBSYSTEM

Default type: Windows 2000/Server 2003 Non-Clustered
 Host Port: 21:00:00:e0:8b:86:ef:b4
 Host Port: 10:00:00:00:c9:71:bf:cd
 Host Port: 21:01:00:e0:8b:a6:5b:9f
 Host Port: 21:00:00:e0:8b:86:5b:9f

Default Group

NVSRAM HOST TYPE DEFINITIONS

HOST TYPE ADT STATUS
 ASSOCIATED INDEX

AIX	Disabled	6
AIX-ADT/AVT	Disabled	4
DEFAULT	Disabled	0
HP-UX	Disabled	7
IBM TS SAN VCE	Disabled	12
Irix	Disabled	10
LNXCLVMWARE	Disabled	13
Linux	Disabled	5
Netware Failover	Disabled	11
Solaris (with Veritas DMP)	Disabled	14
Solaris (with or without MPXIO)	Disabled	8
Unused1	Disabled	1
Windows 2000/Server 2003 Clustered	Disabled	3
Windows 2000/Server 2003 Clustered (supports DMP)	Disabled	15
Windows 2000/Server 2003 Non-Clustered	Disabled	2 (Default)
Windows 2000/Server 2003 Non-Clustered (supports DMP)	Disabled	9

Client Configuration Parameters

Client Registry Parameters

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,\
74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73,\
00,76,00,63,00,68,00,6f,00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00,\
6b,00,20,00,69,00,69,00,73,00,73,00,76,00,63,00,73,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):52,00,50,00,43,00,53,00,53,00,00,00,04,00,54,00,54,00,\
50,00,46,00,69,00,6c,00,74,00,65,00,72,00,00,00,49,00,49,00,53,00,41,00,44,\
00,4d,00,49,00,4e,00,00,00,00,00
```

"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet Information Services Manager"
"FailureActions"=hex:80,51,01,00,01,00,00,00,00,00,00,00,03,00,00,00,53,00,65,\

00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters]

"MajorVersion"=dword:00000006
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINDOWS\\system32\\inetrv"
"AccessDeniedMessage"="Error: Access is Denied."
"ServiceDll"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,44,00,4f,00,57,00,53,\

00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,\

65,00,74,00,73,00,72,00,76,00,5c,00,69,00,69,00,73,00,77,00,33,00,61,00,64,\

00,6d,00,2e,00,64,00,6c,00,6c,00,00,00
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Performance]

"Library"="C:\\WINDOWS\\system32\\inetrv\\w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"PerfIniFile"="w3ctrs.ini"
"Last Counter"=dword:00000a9e
"Last Help"=dword:00000a9f
"First Counter"=dword:000009a8
"First Help"=dword:000009a9
"Object List"="2472 2646"
"Library Validation
Code"=hex:00,db,f6,fe,3d,ba,c6,01,00,5e,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:39,e3,6c,2c,b4,be,59,f5,17,7c,c4,d5,2f,dc,f7,1a
"WbemAdapFileTime"=hex:90,a2,bd,fe,3d,ba,c6,01
"WbemAdapFileSize"=dword:00005e00

"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,00,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\

00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,\

05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\

20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\

00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,\

00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Enum]

"0"="Root\LEGACY_W3SVC\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Parameters]

"ListenBackLog"=dword:00000019
"PoolThreadLimit"=dword:00001ff8
"MaxPoolThreads"=dword:00000ffc
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Performance]

"Library"="infectrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"PerfIniFile"="infectrs.ini"
"Last Counter"=dword:000009a6
"Last Help"=dword:000009a7
"First Counter"=dword:00000966
"First Help"=dword:00000967

"Object List"="2406"
"Library Validation
Code"=hex:00,81,94,fc,3d,ba,c6,01,00,20,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:4c,c3,d3,e7,44,ca,56,e0,f3,e8,a0,14,52,26,fb,0f
"WbemAdapFileTime"=hex:d0,66,76,fb,3d,ba,c6,01
"WbemAdapFileSize"=dword:00002000
"WbemAdapStatus"=dword:00000000
"2003"=hex(b):d4,10,6f,ec,14,c3,c7,01

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\Software\Microsoft\TPCC]

"Path"="C:\\inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:0000000a
"MaxConnections"=dword:0000c350
"MaxPendingDeliveries"=dword:00001388
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="ibmserv4"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
"CallNoDuplicatesNewOrder"=dword:00000001

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client]

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\ConnectTo]

"ibmserv4"="DBMSSOCN,10.0.0.1,1438"
"ibmserv5"="DBMSSOCN,10.0.0.3,1440"
"cp1440"="DBMSSOCN,10.0.0.1,1440"
"cp1453"="DBMSSOCN,10.0.0.1,1453"
"sim78"="DBMSSOCN,10.0.0.1,1478"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0]

"ProtocolsSupported"=hex(7):73,00,6d,00,00,00,74,00,63,00,70,00,00,00,6e,00,70,\
00,00,00,76,00,69,00,61,00,00,00,00,00
"ProtocolOrder"=hex(7):73,00,6d,00,00,00,74,00,63,00,70,00,00,00,6e,00,70,00,\
00,00,00,00

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\GeneralFlags]

"NumberOfFlags"=dword:00000002

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\GeneralFlags\Flag1]

"Label"="Force protocol encryption"
"Value"=dword:00000000

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\GeneralFlags\Flag2]
"Label"="Trust Server Certificate"
"Value"=dword:00000000

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\LastConnect]

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\np]
"DLLname"="SQLNCLI"
"NumberOfFlags"=dword:00000000
"NumberOfProperties"=dword:00000001
"ProtocolName"="Named Pipes"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\np\Property1]
"Name"="Default Pipe"
"Value"="sql\query"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\sm]
"DLLname"="SQLNCLI"
"NumberOfFlags"=dword:00000000
"NumberOfProperties"=dword:00000000
"ProtocolName"="Shared Memory"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\tcp]
"DLLname"="SQLNCLI"
"NumberOfFlags"=dword:00000000
"NumberOfProperties"=dword:00000003
"ProtocolName"="TCP/IP"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\tcp\Property1]
"Name"="Default Port"
"Value"=dword:0000059e

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\tcp\Property2]
"Name"="KEEPALIVE (in milliseconds)"
"Value"=dword:00007530

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\tcp\Property3]
"Name"="KEEPALIVEINTERVAL (in milliseconds)"
"Value"=dword:000003e8

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\via]

"DLLname"="SQLNCLI"
"NumberOfFlags"=dword:00000000
"NumberOfProperties"=dword:00000002
"ProtocolName"="VIA"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\via\Property1]
"Name"="Default Server Port"
"Value"="0:1433"

[HKEY_LOCAL_MACHINE\Software\Microsoft\MSSQLServer\Client\SN19.0\via\Property2]
"Name"="Default Client NIC"
"Value"="0"

Client System Information Report

System Information report written at: 08/28/08 16:02:16
System Name: FCLIENT38
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Standard Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	FCLIENT38
System Manufacturer	IBM
System Model	IBM eServer x3500-[7977AC1]-
System Type	X86-based PC
Processor	x86 Family 6 Model 23 Stepping 6 GenuineIntel ~2992 Mhz
Processor	x86 Family 6 Model 23 Stepping 6 GenuineIntel ~2992 Mhz
Processor	x86 Family 6 Model 23 Stepping 6 GenuineIntel ~2992 Mhz
BIOS Version/Date	IBM IBM BIOS Version 1.64-[SPE164AUS-1.64]-, 4/2/2008
SMBIOS Version	2.34
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)"
User Name	FCLIENT38\Administrator
Time Zone	Eastern Daylight Time
Total Physical Memory	2,046.87 MB
Available Physical Memory	1.73 GB

Total Virtual Memory 3.85 GB
Available Virtual Memory 3.75 GB
Page File Space 2.00 GB
Page File C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
Memory Address 0xC8500000-0xC85FFFFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9
Memory Address 0xC8500000-0xC85FFFFF	Intel(R) PRO/1000 PT Dual Port Server Adapter #5

I/O Port 0x00000000-0x00000CF7	PCI bus
I/O Port 0x00000000-0x00000CF7	Direct memory access controller

I/O Port 0x00000060-0x00000060	Motherboard resources
I/O Port 0x00000060-0x00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard

Memory Address 0xC8700000-0xC87FFFFF	Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 - 2692
--------------------------------------	--

Memory Address 0xC8700000-0xC87FFFFF
Broadcom NetXtreme Gigabit Ethernet #2

Memory Address 0xC8200000-0xC83FFFFF	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
Memory Address 0xC8200000-0xC83FFFFF	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510

Memory Address 0xC8200000-0xC83FFFFF	IBM ServeRAID 8k/8k-l Controller
--------------------------------------	----------------------------------

I/O Port 0x00000064-0x00000064	Motherboard resources
I/O Port 0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard

IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2688
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2689
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268A
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268B
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB2

Enhanced Host Controller - 268C

I/O Port 0x00003000-0x00003FFF Intel(R) 5000 Series
Chipset PCI Express x8 Port 2-3 - 25F7
I/O Port 0x00003000-0x00003FFF Intel(R)
6311ESB/6321ESB PCI Express Upstream Port - 3500
I/O Port 0x00003000-0x00003FFF Intel(R)
6311ESB/6321ESB PCI Express Downstream Port E1 - 3510

I/O Port 0x00003000-0x00003FFF IBM ServeRAID
8k/8k-l Controller

Memory Address 0xC8B00000-0xC8BFFFFF Intel(R)
5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
Memory Address 0xC8B00000-0xC8BFFFFF Intel(R)
6311ESB/6321ESB PCI Express Upstream Port - 3500
Memory Address 0xC8B00000-0xC8BFFFFF Intel(R)
6311ESB/6321ESB PCI Express Downstream Port E1 - 3510

Memory Address 0xC8B00000-0xC8BFFFFF IBM
ServeRAID 8k/8k-l Controller

IRQ 16 Intel(R) 5000P Chipset Memory Controller Hub -
25D8

IRQ 16 Intel(R) 6311ESB/6321ESB PCI Express
Downstream Port E2 - 3514

IRQ 16 Intel(R) 5000 Series Chipset PCI Express x8 Port 4-5
- 25F8

IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 5 -
25E5

IRQ 16 Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7
- 25F9

IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 7 -
25E7

IRQ 16 Intel(R) 631xESB/6321ESB/3100 Chipset PCI
Express Root Port 1 - 2690

IRQ 16 Broadcom NetXtreme Gigabit Ethernet

IRQ 16 ATI ES1000

IRQ 17 Intel(R) PRO/1000 PT Dual Port Server Adapter #7

IRQ 17 Intel(R) PRO/1000 PT Dual Port Server Adapter #5

IRQ 17 Intel(R) 631xESB/6321ESB/3100 Chipset PCI
Express Root Port 2 - 2692

IRQ 17 Broadcom NetXtreme Gigabit Ethernet #2

IRQ 18 Intel(R) PRO/1000 PT Dual Port Server Adapter #8

IRQ 18 Intel(R) PRO/1000 PT Dual Port Server Adapter #6

IRQ 19 Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3
- 25F7

IRQ 19 Intel(R) 6311ESB/6321ESB PCI Express Upstream
Port - 3500

IRQ 19 Intel(R) 6311ESB/6321ESB PCI Express
Downstream Port E1 - 3510

IRQ 19 IBM ServeRAID 8k/8k-l Controller

IRQ 19 Intel(R) 5000 Series Chipset PCI Express x4 Port 3 -
25E3

Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF ATI ES1000

Memory Address 0xC8600000-0xC86FFFFF Intel(R)
631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 -
2690

Memory Address 0xC8600000-0xC86FFFFF
Broadcom NetXtreme Gigabit Ethernet

Memory Address 0xC8400000-0xC84FFFFF Intel(R)
5000 Series Chipset PCI Express x8 Port 4-5 - 25F8

Memory Address 0xC8400000-0xC84FFFFF Intel(R)
PRO/1000 PT Dual Port Server Adapter #7

[DMA]

Resource	Device	Status
Channel 4	Direct memory access controller	OK
Channel 1	ECP Printer Port (LPT1)	OK

[Forced Hardware]

Device	PNP Device ID
--------	---------------

[I/O]

Resource	Device	Status
0x00000000-0x00000CF7	PCI bus	OK
0x00000000-0x00000CF7	Direct memory access controller	OK
0x00000D00-0x0000FFFF	PCI bus	OK
0x00003000-0x00003FFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	OK
0x00003000-0x00003FFF	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500	OK
0x00003000-0x00003FFF	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510	OK
0x00003000-0x00003FFF	IBM ServeRAID 8k/8k-l Controller	OK
0x00004000-0x00004FFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 4-5 - 25F8	OK
0x00005000-0x00005FFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9	OK
0x00001800-0x0000181F	Intel(R)	

631xESB/6321ESB/3100 Chipset USB Universal Host
Controller - 2688 OK
0x00001820-0x0000183F Intel(R)
631xESB/6321ESB/3100 Chipset USB Universal Host
Controller - 2689 OK
0x00001840-0x0000185F Intel(R)
631xESB/6321ESB/3100 Chipset USB Universal Host
Controller - 268A OK
0x00001860-0x0000187F Intel(R)
631xESB/6321ESB/3100 Chipset USB Universal Host
Controller - 268B OK
0x00006000-0x000060FF ATI ES1000 OK

0x000003B0-0x000003BB ATI ES1000 OK

0x000003C0-0x000003DF ATI ES1000 OK

0x00000A79-0x00000A79 ISAPNP Read Data Port
OK

0x00000279-0x00000279 ISAPNP Read Data Port
OK

0x00000274-0x00000277 ISAPNP Read Data Port
OK

0x00000010-0x0000001F Motherboard resources
OK

0x00000024-0x00000025 Motherboard resources
OK

0x00000028-0x00000029 Motherboard resources
OK

0x0000002C-0x0000002D Motherboard resources
OK

0x0000002E-0x0000002F Motherboard resources
OK

0x00000030-0x00000031 Motherboard resources
OK

0x00000034-0x00000035 Motherboard resources
OK

0x00000038-0x00000039 Motherboard resources
OK

0x0000003C-0x0000003D Motherboard resources
OK

0x0000004E-0x0000004F Motherboard resources
OK

0x00000050-0x00000053 Motherboard resources
OK

0x00000060-0x00000060 Motherboard resources
OK

0x00000060-0x00000060 Standard 101/102-Key or
Microsoft Natural PS/2 Keyboard OK

0x00000062-0x00000062 Motherboard resources
OK

0x00000063-0x00000063 Motherboard resources
OK

0x00000064-0x00000064 Motherboard resources

OK	
0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
0x00000065-0x00000065	Motherboard resources OK
0x00000066-0x00000066	Motherboard resources OK
0x00000067-0x00000067	Motherboard resources OK
0x00000072-0x00000077	Motherboard resources OK
0x00000080-0x00000080	Motherboard resources OK
0x00000090-0x0000009F	Motherboard resources OK
0x000000A4-0x000000A5	Motherboard resources OK
0x000000A8-0x000000A9	Motherboard resources OK
0x000000AC-0x000000AD	Motherboard resources OK
0x000000B0-0x000000B5	Motherboard resources OK
0x000000B8-0x000000B9	Motherboard resources OK
0x000000BC-0x000000BD	Motherboard resources OK
0x000004D0-0x000004D1	Motherboard resources OK
0x00001000-0x0000107F	Motherboard resources OK
0x00001180-0x000011BF	Motherboard resources OK
0x0000FE00-0x0000FE00	Motherboard resources OK
0x00000800-0x0000083F	Motherboard resources OK
0x00000600-0x0000063F	Motherboard resources OK
0x00000CA8-0x00000CAB	Motherboard resources OK
0x00000CAC-0x00000CAF	Motherboard resources OK
0x00000081-0x00000091	Direct memory access controller OK
0x00000093-0x0000009F	Direct memory access controller OK
0x000000C0-0x000000DF	Direct memory access controller OK
0x000000F0-0x000000FE	Numeric data processor OK
0x00000020-0x00000021	Programmable interrupt controller OK
0x000000A0-0x000000A1	Programmable interrupt controller OK

0x00000070-0x00000071	System CMOS/real time clock OK
0x00000061-0x00000061	System speaker OK
0x00000040-0x00000043	System timer OK
0x000003F8-0x000003FF	Communications Port (COM1) OK
0x00000378-0x0000037F	ECP Printer Port (LPT1) OK
0x00000778-0x0000077F	ECP Printer Port (LPT1) OK
0x000002F8-0x000002FF	Communications Port (COM2) OK
0x000018C0-0x000018CF	Intel(R) 631xESB/6321ESB Ultra ATA Storage Controller - 269E OK
0x000001F0-0x000001F7	Primary IDE Channel OK
0x000003F6-0x000003F6	Primary IDE Channel OK
0x00002080-0x0000209F	Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B OK
[IRQs]	
Resource	Device Status
IRQ 9	Microsoft ACPI-Compliant System OK
IRQ 16	Intel(R) 5000P Chipset Memory Controller Hub - 25D8 OK
IRQ 16	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E2 - 3514 OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x8 Port 4-5 - 25F8 OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5 OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9 OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7 OK
IRQ 16	Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 - 2690 OK
IRQ 16	Broadcom NetXtreme Gigabit Ethernet OK
IRQ 16	ATI ES1000 OK
IRQ 19	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7 OK
IRQ 19	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500 OK
IRQ 19	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510 OK
IRQ 19	IBM ServeRAID 8k/8k-l Controller OK

IRQ 19	Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3 OK
IRQ 17	Intel(R) PRO/1000 PT Dual Port Server Adapter #7 OK
IRQ 17	Intel(R) PRO/1000 PT Dual Port Server Adapter #5 OK
IRQ 17	Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 - 2692 OK
IRQ 17	Broadcom NetXtreme Gigabit Ethernet #2 OK
IRQ 18	Intel(R) PRO/1000 PT Dual Port Server Adapter #8 OK
IRQ 18	Intel(R) PRO/1000 PT Dual Port Server Adapter #6 OK
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2688 OK
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2689 OK
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268A OK
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268B OK
IRQ 23	Intel(R) 631xESB/6321ESB/3100 Chipset USB Enhanced Host Controller - 268C OK
IRQ 13	Numeric data processor OK
IRQ 8	System CMOS/real time clock OK
IRQ 0	System timer OK
IRQ 1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
IRQ 12	PS/2 Compatible Mouse OK
IRQ 4	Communications Port (COM1) OK
IRQ 3	Communications Port (COM2) OK
IRQ 14	Primary IDE Channel OK
IRQ 10	Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	ATI ES1000	OK
0xD4000-0xD7FFF	PCI bus	OK
0xD8000-0xDBFFF	PCI bus	OK
0xDC000-0xDFFFF	PCI bus	OK
0x80000000-0xFDFDFDF	PCI bus	OK
0xC8100000-0xC83FFFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	OK
0xC8B00000-0xC8BFFFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	OK
0xC8B00000-0xC8BFFFF	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500	OK
0xC8B00000-0xC8BFFFF	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510	OK

0xC8B00000-0xC8BFFFFF IBM ServeRAID 8k/8k-1
Controller OK

0xC8200000-0xC83FFFFF Intel(R) 6311ESB/6321ESB
PCI Express Upstream Port - 3500 OK

0xC8200000-0xC83FFFFF Intel(R) 6311ESB/6321ESB
PCI Express Downstream Port E1 - 3510 OK

0xC8200000-0xC83FFFFF IBM ServeRAID 8k/8k-1
Controller OK

0xC8400000-0xC84FFFFF Intel(R) 5000 Series Chipset
PCI Express x8 Port 4-5 - 25F8 OK

0xC8400000-0xC84FFFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #7 OK

0xC8420000-0xC843FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #7 OK

0xC8460000-0xC847FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #8 OK

0xC8440000-0xC845FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #8 OK

0xC8500000-0xC85FFFFF Intel(R) 5000 Series Chipset
PCI Express x8 Port 6-7 - 25F9 OK

0xC8500000-0xC85FFFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #5 OK

0xC8520000-0xC853FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #5 OK

0xC8560000-0xC857FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #6 OK

0xC8540000-0xC855FFFF Intel(R) PRO/1000 PT Dual
Port Server Adapter #6 OK

0xC8600000-0xC86FFFFF Intel(R)
631xESB/6321ESB/3100 Chipset PCI Express Root Port 1 -
2690 OK

0xC8600000-0xC86FFFFF Broadcom NetXtreme Gigabit
Ethernet OK

0xC8700000-0xC87FFFFF Intel(R)
631xESB/6321ESB/3100 Chipset PCI Express Root Port 2 -
2692 OK

0xC8700000-0xC87FFFFF Broadcom NetXtreme Gigabit
Ethernet #2 OK

0xFDFFFC00-0xFDFFFFFF Intel(R)
631xESB/6321ESB/3100 Chipset USB2 Enhanced Host
Controller - 268C OK

0xD0000000-0xD7FFFFFF ATI ES1000 OK

0xC8800000-0xC880FFFF ATI ES1000 OK

0xE0000000-0xEFFFFFFF Motherboard resources
OK

0xFEE00000-0xFEE0FFFF Motherboard resources
OK

0xFEC81000-0xFEC81FFF Motherboard resources
OK

0xFEC81400-0xFEC823FF Motherboard resources
OK

0xC8000000-0xC800FFFF Motherboard resources

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\SL_ANET.ACM	3.02	84.00 KB (86,016 bytes)	3/25/2003 8:00 AM
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)	3/25/2003 8:00 AM
c:\windows\system32\tsoft32.acm	DSP GROUP, INC.		OK
C:\WINDOWS\system32\TSSOFT32.ACM	1.01	9.50 KB (9,728 bytes)	3/25/2003 8:00 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)	3/25/2003 8:00 AM
c:\windows\system32\msg711.acm	Microsoft Corporation		OK
C:\WINDOWS\system32\MSG711.ACM	5.2.3790.0 (srv03_rtm.030324-2048)	10.00 KB (10,240 bytes)	3/25/2003 8:00 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)	3/25/2003 8:00 AM
c:\windows\system32\l3codeca.acm	Fraunhofer Institut Integrierte Schaltungen IIS	Fraunhofer IIS MPEG Layer-3 Codec	OK
C:\WINDOWS\system32\L3CODECA.ACM	1, 9, 0, 0305	284.00 KB (290,816 bytes)	3/25/2003 8:00 AM
c:\windows\system32\msg723.acm	Microsoft Corporation		OK
C:\WINDOWS\system32\MSG723.ACM	5.2.3790.1830	120.00 KB (122,880 bytes)	

8/7/2006 1:38 PM

c:\windows\system32\imaadp32.acm Microsoft Corporation OK

C:\WINDOWS\system32\IMAADP32.ACM

5.2.3790.0 (srv03_rtm.030324-2048) 15.50 KB (15,872 bytes) 3/25/2003 8:00 AM

[Video Codecs]

CODEC	Manufacturer	Description	Status
File	Version	Size	Creation Date
c:\windows\system32\iyuv_32.dll	Microsoft Corporation		OK
C:\WINDOWS\system32\IYUV_32.DLL	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	46.50 KB (47,616 bytes)	8/7/2006 1:38 PM
c:\windows\system32\msh261.drv	Microsoft Corporation		OK
C:\WINDOWS\system32\MSH261.DRV	5.2.3790.1830	184.00 KB (188,416 bytes)	8/7/2006 1:38 PM
c:\windows\system32\msyuv.dll	Microsoft Corporation		OK
C:\WINDOWS\system32\MSYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	16.50 KB (16,896 bytes)	3/24/2003 9:49 PM
c:\windows\system32\msvidc32.dll	Microsoft Corporation		OK
C:\WINDOWS\system32\MSVIDC32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	26.50 KB (27,136 bytes)	3/25/2003 8:00 AM
c:\windows\system32\msh263.drv	Microsoft Corporation		OK
C:\WINDOWS\system32\MSH263.DRV	5.2.3790.1830	288.00 KB (294,912 bytes)	8/7/2006 1:38 PM
c:\windows\system32\msrle32.dll	Microsoft Corporation		OK
C:\WINDOWS\system32\MSRLE32.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	10.50 KB (10,752 bytes)	3/25/2003 8:00 AM
c:\windows\system32\tsbyuv.dll	Microsoft Corporation		OK
C:\WINDOWS\system32\TSBYUV.DLL	5.2.3790.0 (srv03_rtm.030324-2048)	8.00 KB (8,192 bytes)	3/24/2003 9:50 PM

[CD-ROM]

Item	Value
Drive D:	
Description	CD-ROM Drive

Media Loaded No
 Media Type CD-ROM
 Name TSSTcorp DVD-ROM TS-H352D
 Manufacturer (Standard CD-ROM drives)
 Status OK
 Transfer Rate Not Available
 SCSI Target ID 0
 PNP Device ID IDE\CDROMTSSTCORP_DVD-ROM_TS-H352D\LE01\5&3B7DE513&0&0.0.0

Driver c:\windows\system32\drivers\cdrom.sys
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 51.00 KB
 (52,224 bytes), 3/25/2003 8:00 AM)

[Sound Device]

Item	Value
------	-------

[Display]

Item	Value
Name	ATI ES1000
PNP Device ID	PCI\VEN_1002&DEV_515E&SUBSYS_03051014&REV_02\4&2014205D&0&20F0
Adapter Type	ATI ES1000 (0x515E), ATI Technologies Inc. compatible
Adapter Description	ATI ES1000
Adapter RAM	16.00 MB (16,777,216 bytes)
Installed Drivers	ati2dvag.dll
Driver Version	6.14.10.6583
INF File	oem1.inf (ati2mtag_RN50 section)
Color Planes	1
Color Table Entries	65536
Resolution	1024 x 768 x 75 hertz
Bits/Pixel	16
Memory Address	0xD0000000-0xD7FFFFFF
I/O Port	0x00006000-0x000060FF
Memory Address	0xC8800000-0xC880FFFF
IRQ Channel	IRQ 16
I/O Port	0x000003B0-0x000003BB
I/O Port	0x000003C0-0x000003DF
Memory Address	0xA0000-0xBFFFF
Driver	c:\windows\system32\drivers\ati2mtag.sys (6.14.10.6583, 1.32 MB (1,379,328 bytes), 12/6/2005 9:44 PM)

[Infrared]

Item	Value
------	-------

[Input]

[Keyboard]

Item	Value
Description	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name	Enhanced (101- or 102-key)
Layout	00000409
PNP Device ID	ACPI\PNP0303\4&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
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 54.50 KB
 (55,808 bytes), 3/25/2003 8:00 AM)

[Pointing Device]

Item	Value
Hardware Type	PS/2 Compatible Mouse
Number of Buttons	3
Status	OK
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
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 54.50 KB
 (55,808 bytes), 3/25/2003 8:00 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	8/28/2008 2:38 PM
Index	1
Service Name	AsyncMac

IP Address Not Available
 IP Subnet Not Available
 Default IP Gateway Not Available
 DHCP Enabled No
 DHCP Server Not Available
 DHCP Lease Expires Not Available
 DHCP Lease Obtained Not Available
 MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
 Adapter Type Not Available
 Product Type WAN Miniport (L2TP)
 Installed Yes
 PNP Device ID ROOT\MS_L2TPMINIPOINT\0000

Last Reset 8/28/2008 2:38 PM

Index	Value
2	Service Name Rasl2tp
	IP Address Not Available
	IP Subnet Not Available
	Default IP Gateway Not Available
	DHCP Enabled No
	DHCP Server Not Available
	DHCP Lease Expires Not Available
	DHCP Lease Obtained Not Available
	MAC Address Not Available
	Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 66.00 KB (67,584 bytes), 3/25/2003 8:00 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 8/28/2008 2:38 PM

Index	Value
3	Service Name PptpMiniport
	IP Address Not Available
	IP Subnet Not Available
	Default IP Gateway Not Available
	DHCP Enabled No
	DHCP Server Not Available
	DHCP Lease Expires Not Available
	DHCP Lease Obtained Not Available
	MAC Address 50:50:54:50:30:30
	Driver c:\windows\system32\drivers\raspptp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 61.00 KB (62,464 bytes), 3/25/2003 8:00 AM)

Name [00000004] WAN Miniport (PPPOE)
 Adapter Type Wide Area Network (WAN)
 Product Type WAN Miniport (PPPOE)

Installed Yes
PNP Device ID ROOT\MS_PPPOEMINI\PORT\0000

Last Reset 8/28/2008 2:38 PM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\rasppoe.sys
(5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 40.00 KB
(40,960 bytes), 3/25/2003 8:00 AM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINI\PORT\0000

Last Reset 8/28/2008 2:38 PM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys
(5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 19.50 KB
(19,968 bytes), 3/25/2003 8:00 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 8/28/2008 2:38 PM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available

MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys
(5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 91.00 KB
(93,184 bytes), 3/25/2003 8:00 AM)

Name [00000007] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet

Installed Yes
PNP Device ID
PCI\VEN_14E4&DEV_1659&SUBSYS_02C61014
&REV_214&187919FE&0&00E0
Last Reset 8/28/2008 2:38 PM
Index 7
Service Name b57w2k
IP Address 192.168.40.38
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:1A:64:A1:86:08
Memory Address 0xC8600000-0xC86FFFFF
IRQ Channel IRQ 16
Driver c:\windows\system32\drivers\b57xp32.sys (10.24.0.0
built by: WinDDK, 156.50 KB (160,256 bytes), 8/7/2006 12:45
PM)

Name [00000008] Broadcom NetXtreme Gigabit Ethernet
Adapter Type Ethernet 802.3
Product Type Broadcom NetXtreme Gigabit Ethernet

Installed Yes
PNP Device ID
PCI\VEN_14E4&DEV_1659&SUBSYS_02C61014
&REV_214&110C88BD&0&00E1
Last Reset 8/28/2008 2:38 PM
Index 8
Service Name b57w2k
IP Address 0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:1A:64:A1:86:09
Memory Address 0xC8700000-0xC87FFFFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\b57xp32.sys (10.24.0.0

built by: WinDDK, 156.50 KB (160,256 bytes), 8/7/2006 12:45
PM)

Name [00000009] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:38 PM
Index 9
Service Name e1express
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000010] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:38 PM
Index 10
Service Name e1express
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000011] Intel(R) PRO/1000 PT Dual Port Server
Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server
Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:38 PM
Index 11
Service Name e1express
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available

DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000012] Intel(R) PRO/1000 PT Dual Port Server Adapter
Adapter Type Not Available
Product Type Intel(R) PRO/1000 PT Dual Port Server Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 8/28/2008 2:38 PM
Index 12
Service Name e1express
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000013] Intel(R) PRO/1000 PT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Dual Port Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_064&30A54032&0&0030
Last Reset 8/28/2008 2:38 PM
Index 13
Service Name e1express
IP Address 192.168.102.99
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:17:6D:03:DE
Memory Address 0xC8520000-0xC853FFFF
Memory Address 0xC8500000-0xC85FFFFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\le1e5132.sys (9.6.31.0 built by: WinDDK, 240.90 KB (246,680 bytes), 11/1/2006 11:39 AM)

Name [00000014] Intel(R) PRO/1000 PT Dual Port Server Adapter

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Dual Port Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_064&30A54032&0&0130
Last Reset 8/28/2008 2:38 PM
Index 14
Service Name e1express
IP Address 10.0.0.38
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:17:6D:03:DF
Memory Address 0xC8560000-0xC857FFFF
Memory Address 0xC8540000-0xC855FFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\le1e5132.sys (9.6.31.0 built by: WinDDK, 240.90 KB (246,680 bytes), 11/1/2006 11:39 AM)

Name [00000015] Intel(R) PRO/1000 PT Dual Port Server Adapter
Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Dual Port Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_064&8E1D94C&0&0020
Last Reset 8/28/2008 2:38 PM
Index 15
Service Name e1express
IP Address 192.168.100.99
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:17:6D:02:2C
Memory Address 0xC8420000-0xC843FFFF
Memory Address 0xC8400000-0xC84FFFFF
IRQ Channel IRQ 17
Driver c:\windows\system32\drivers\le1e5132.sys (9.6.31.0 built by: WinDDK, 240.90 KB (246,680 bytes), 11/1/2006 11:39 AM)

Name [00000016] Intel(R) PRO/1000 PT Dual Port Server Adapter

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 PT Dual Port Server Adapter
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_064&8E1D94C&0&0120
Last Reset 8/28/2008 2:38 PM
Index 16
Service Name e1express
IP Address 192.168.101.99
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:17:6D:02:2D
Memory Address 0xC8460000-0xC847FFFF
Memory Address 0xC8440000-0xC845FFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\le1e5132.sys (9.6.31.0 built by: WinDDK, 240.90 KB (246,680 bytes), 11/1/2006 11:39 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 No

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_{3F69A108-AF83-496B-B678-67D782783069}] SEQPACKET 11
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_{3F69A108-AF83-496B-B678-67D782783069}] DATAGRAM 11
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_{0F6BF416-E2F3-42EC-B407-0805CD3A677D}] SEQPACKET 10
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_{0F6BF416-E2F3-42EC-B407-0805CD3A677D}] DATAGRAM 10
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_{3885E7C4-14DE-4B94-AA49-BFE7EF83C2F6}] SEQPACKET 9
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_{3885E7C4-14DE-4B94-AA49-BFE7EF83C2F6}] DATAGRAM 9
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_{5DAEEA78-0B36-4DE7-ADE2-D1046DCD6948}] SEQUENCE 8
 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_{5DAEEA78-0B36-4DE7-ADE2-D1046DCD6948}] DATAGRAM 8
 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_{C3ADAAAF3-F470-4C67-BF36-939550D99734}] SEQUENCE 7
 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_{C3ADAAAF3-F470-4C67-BF36-939550D99734}] DATAGRAM 7
 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_{CE6D789D-D708-4CF8-A83B-40CB07D1860F}] SEQUENCE 6
 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_{CE6D789D-D708-4CF8-A83B-40CB07D1860F}] DATAGRAM 6
 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_{353FE206-6718-45C5-8007-F35869C12051}] SEQUENCE 5
 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_{353FE206-6718-45C5-8007-F35869C12051}] DATAGRAM 5

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_{4054C2E5-1692-44B4-AD9C-974BCB075979}] SEQPACKE T 4
 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_{4054C2E5-1692-44B4-AD9C-974BCB075979}] DATAGRAM 4
 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_{220FB7B8-ACA3-4E5F-A690-DF26F03159FA}] SEQPACKE T 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_{220FB7B8-ACA3-4E5F-A690-DF26F03159FA}] 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_{97AFF47B-14F6-4250-B94B-93CAB8B87CCF}] SEQPACKE T 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_{97AFF47B-14F6-4250-B94B-93CAB8B87CCF}] 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_{88A93A75-19C0-4C58-ADB1-C14E5B49EB57}] SEQPACKE T 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_{88A93A75-19C0-4C58-ADB1-C14E5B49EB57}] 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_{45AEBD6E-2CEE-4126-8530-4C726FC4C378}] SEQUENCE 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_{45AEBD6E-2CEE-4126-8530-4C726FC4C378}] 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 (COM1)
Status	OK
PNP Device ID	ACPI\PNP0501\1
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 0x000003F8-0x000003FF
 IRQ Channel IRQ 4
 Driver c:\windows\system32\drivers\serial.sys
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 64.00 KB
 (65,536 bytes), 3/25/2003 8:00 AM)

Name Communications Port (COM2)
 Status OK
 PNP Device ID ACPI\PNP0501\2
 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
 (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 64.00 KB
 (65,536 bytes), 3/25/2003 8:00 AM)

[Parallel]

Item	Value
Name	LPT1
PNP Device ID	ACPI\PNP0401\4
I/O Port	0x00000378-0x0000037F
I/O Port	0x00000778-0x0000077F
DMA Channel	Channel 1
Driver	c:\windows\system32\drivers\parport.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 79.50 KB (81,408 bytes), 3/24/2003 7:04 PM)

[Storage]

[Drives]

Item	Value
Drive	C:
Description	Local Fixed Disk
Compressed	No
File System	NTFS
Size	68.24 GB (73,270,759,424 bytes)
Free Space	59.90 GB (64,317,763,584 bytes)
Volume Name	
Volume Serial Number	0020B154

Drive	D:
Description	CD-ROM Disc

Drive	X:
Description	Network Connection
Provider Name	\\fsserv\ray

Drive	Y:
Description	Network Connection
Provider Name	\\fsserv\edrive

[Disks]

Item	Value
Description	Disk drive
Manufacturer	(Standard disk drives)
Model	Adaptec Array SCSI Disk Device
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	1
SCSI Bus	4
SCSI Logical Unit	0
SCSI Port	1
SCSI Target ID	0
Sectors/Track	63
Size	136.61 GB (146,681,418,240 bytes)
Total Cylinders	17,833
Total Sectors	286,487,145
Total Tracks	4,547,415
Tracks/Cylinder	255
Partition	Disk #0, Partition #0
Partition Size	68.24 GB (73,270,761,984 bytes)
Partition Starting Offset	32,256 bytes

[SCSI]

Item	Value
Name	IBM ServeRAID 8k/8k-l Controller
Manufacturer	Adaptec
Status	OK
PNP Device ID	PCI\VEN_9005&DEV_0286&SUBSYS_95801014& REV_02\6&305972A8&0&00000010
Memory Address	0xC8200000-0xC83FFFFF
Memory Address	0xC8B00000-0xC8BFFFFF
I/O Port	0x00003000-0x00003FFF
IRQ Channel	IRQ 19
Driver	c:\windows\system32\drivers\arcasas.sys (5.1.0.9206 built by: WinDDK, 50.00 KB (51,200 bytes), 8/4/2006 2:38 PM)

[IDE]

Item	Value
Name	Intel(R) 631xESB/6321ESB Ultra ATA Storage Controller - 269E
Manufacturer	Intel
Status	OK
PNP Device ID	PCI\VEN_8086&DEV_269E&SUBSYS_03321014& REV_09\3&61AAA01&0&F9
I/O Port	0x000018C0-0x000018CF
Driver	c:\windows\system32\drivers\pciide.sys (5.2.3790.0 (srv03_rtm.030324-2048), 5.50 KB (5,632 bytes), 8/7/2006 12:33 PM)

Name	Primary IDE Channel
Manufacturer	(Standard IDE ATA/ATAPI controllers)

Status	OK
PNP Device ID	PCI\IDE\IDECHANNEL\4&379AE00F&0&0

I/O Port	0x000001F0-0x000001F7
I/O Port	0x000003F6-0x000003F6
IRQ Channel	IRQ 14
Driver	c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 8/7/2006 12:33 PM)

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code
--------	---------------	------------

[USB]

Device	PNP Device ID
Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2688	PCI\VEN_8086&DEV_2688&SUBSYS_03321014& REV_09\3&61AAA01&0&E8
Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 2689	PCI\VEN_8086&DEV_2689&SUBSYS_03321014& REV_09\3&61AAA01&0&E9
Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268A	PCI\VEN_8086&DEV_268A&SUBSYS_03321014 &REV_09\3&61AAA01&0&EA
Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host Controller - 268B	PCI\VEN_8086&DEV_268B&SUBSYS_03321014& REV_09\3&61AAA01&0&EB
Intel(R) 631xESB/6321ESB/3100 Chipset USB2 Enhanced Host Controller - 268C	PCI\VEN_8086&DEV_268C&SUBSYS_03321014& REV_09\3&61AAA01&0&EF

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started
------	-------------	------	------	---------

	Start Mode	State	Status	Error Control	
	Accept	Pause	Accept	Stop	
abiosdsk	Abiosdsk	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Ignore
acpi	Microsoft ACPI Driver				
Driver	c:\windows\system32\drivers\acpi.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			
acpiec	ACPIEC	c:\windows\system32\drivers\acpiec.sys			
	Kernel Driver	No	Disabled	Stopped	
	OK	Normal	No	No	
adpu160m	adpu160m	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
adpu320	adpu320	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
afcnt	afcnt	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
afd	AFD Networking Support Environment				
Driver	c:\windows\system32\drivers\afd.sys			Kernel	
	Yes	System	Running	OK	Normal
	No	Yes			
aha154x	Aha154x	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
aic78u2	aic78u2	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
aic78xx	aic78xx	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
aliide	Alilide	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
arcsas	Adaptec SAS RAID Storport's Miniport Driver				
Driver	c:\windows\system32\drivers\arcsas.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			
asynmac	RAS Asynchronous Media Driver				
Driver	c:\windows\system32\drivers\asynmac.sys			Kernel	
	No	Manual	Stopped	OK	Normal
	No	No			
atapi	Standard IDE/ESDI Hard Disk Controller				
Driver	c:\windows\system32\drivers\atapi.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			
atdisk	Atdisk	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Ignore
	No	No			
ati2mtag	ati2mtag	c:\windows\system32\drivers\ati2mtag.sys			
	Kernel Driver	Yes	Manual	Running	

	OK	Ignore	No	Yes	
atmarpc	ATM ARP Client Protocol				
Driver	c:\windows\system32\drivers\atmarpc.sys			Kernel	
	No	Manual	Stopped	OK	Normal
	No	No			
audstub	Audio Stub Driver				
Driver	c:\windows\system32\drivers\audstub.sys			Kernel	
	Yes	Manual	Running	OK	Normal
	No	Yes			
b57w2k	Broadcom NetXtreme Gigabit Ethernet				
Driver	c:\windows\system32\drivers\b57xp32.sys			Kernel	
	Yes	Manual	Running	OK	Normal
	No	Yes			
beep	Beep	c:\windows\system32\drivers\beep.sys			
	Kernel Driver	Yes	System	Running	
	OK	Normal	No	Yes	
cbidf2k	cbidf2k	c:\windows\system32\drivers\cbidf2k.sys			
	Kernel Driver	No	Disabled	Stopped	
	OK	Normal	No	No	
cd20xrmt	cd20xrmt	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cdfs	Cdfs	c:\windows\system32\drivers\cdfs.sys			
	File System Driver	Yes	Disabled	Running	
	OK	Normal	No	Yes	
cdrom	CD-ROM Driver				
Driver	c:\windows\system32\drivers\cdrom.sys			Kernel	
	Yes	System	Running	OK	Normal
	No	Yes			
changer	Changer	Not Available		Kernel Driver	
	No	System	Stopped	OK	Ignore
	No	No			
clusdisk	Cluster Disk Driver				
Driver	c:\windows\system32\drivers\clusdisk.sys			Kernel	
	No	Disabled	Stopped	OK	Normal
	No	No			
cmdide	CmdIde	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cpqarray	Cpqarray	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cpqarray2	cpqarray2	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cpqcissm	cpqcissm	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cpqfcalm	cpqfcalm	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
cpuz129	cpuz129				
	Kernel Driver	Yes	Manual	Running	
ys	Kernel Driver	No	Manual	Stopped	

	OK	Normal	No	No	
credisk	CRC Disk Filter Driver				
Driver	c:\windows\system32\drivers\credisk.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			
dac960nt	dac960nt	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
dellcerc	dellcerc	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
dfsdriver	DfsDriver	c:\windows\system32\drivers\dfs.sys			
	File System Driver	Yes	Boot	Running	
	OK	Normal	No	Yes	
disk	Disk Driver				
Driver	c:\windows\system32\drivers\disk.sys			Kernel	
	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				
Driver	c:\windows\system32\drivers\dmio.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			
dmload	dmload	c:\windows\system32\drivers\dmload.sys			
	Kernel Driver	Yes	Boot	Running	
	OK	Normal	No	Yes	
dpti2o	dpti2o	Not Available		Kernel Driver	
	No	Disabled	Stopped	OK	Normal
	No	No			
e1express	Intel(R) PRO/1000 PCI Express Network Connection				
Driver	c:\windows\system32\drivers\ele5132.sys			Kernel	
	Yes	Manual	Running	OK	Normal
	No	Yes			
fastfat	Fastfat	c:\windows\system32\drivers\fastfat.sys			
	File System Driver	No	Disabled	Stopped	
	OK	Normal	No	No	
fdc	Fdc	c:\windows\system32\drivers\fdc.sys			
	Kernel Driver	No	System	Stopped	
	OK	Ignore	No	No	
fips	Fips	c:\windows\system32\drivers\fips.sys			
	Kernel Driver	Yes	System	Running	
	OK	Normal	No	Yes	
flpydisk	Flpydisk	c:\windows\system32\drivers\flpydisk.sys			
	Kernel Driver	No	System	Stopped	
	OK	Ignore	No	No	
ftmgr	FltMgr	c:\windows\system32\drivers\ftmgr.sys			
	File System Driver	Yes	Boot	Running	
	OK	Normal	No	Yes	
ftdisk	Volume Manager Driver				
Driver	c:\windows\system32\drivers\ftdisk.sys			Kernel	
	Yes	Boot	Running	OK	Normal
	No	Yes			

gpc	Generic Packet Classifier					
Driver	c:\windows\system32\drivers\msgpc.sys	Kernel	Yes	Manual	Running	OK Normal
hpn	hpn	Kernel Driver	No	Yes		
	hpn	Not Available	No	Disabled	Stopped	OK Normal
hpt3xx	hpt3xx	Kernel Driver	No	No		
	hpt3xx	Not Available	No	Disabled	Stopped	OK Normal
http	HTTP	c:\windows\system32\drivers\http.sys	Kernel Driver	Yes	Manual	Running
	http	Kernel Driver	Yes	Manual	Running	OK Normal
i2omgmt	i2omgmt	Kernel Driver	No	System	Stopped	OK Normal
	i2omgmt	Not Available	No	System	Stopped	OK Normal
i2omp	i2omp	Kernel Driver	No	Disabled	Stopped	OK Normal
	i2omp	Not Available	No	Disabled	Stopped	OK Normal
i804prt	i804 Keyboard and PS/2 Mouse Port Driver					
Driver	c:\windows\system32\drivers\i804prt.sys	Kernel	Yes	System	Running	OK Normal
iirsp	iirsp	Kernel Driver	No	Yes		
	iirsp	Not Available	No	Disabled	Stopped	OK Normal
imapi	CD-Burning Filter Driver					
Driver	c:\windows\system32\drivers\imapi.sys	Kernel	No	System	Stopped	OK Normal
intelide	IntelIde	Kernel Driver	No	No		
	intelide	Not Available	No	Disabled	Stopped	OK Normal
intelppm	Intel Processor Driver					
Driver	c:\windows\system32\drivers\intelppm.sys	Kernel	Yes	Manual	Running	OK Normal
ip6fw	IPv6 Windows Firewall Driver					
Driver	c:\windows\system32\drivers\ip6fw.sys	Kernel	No	Manual	Stopped	OK Normal
ipfilterdriver	IP Traffic Filter Driver					
Driver	c:\windows\system32\drivers\ipfltdrv.sys	Kernel	No	Manual	Stopped	OK Normal
ipinip	IP in IP Tunnel Driver					
Driver	c:\windows\system32\drivers\ipinip.sys	Kernel	No	Manual	Stopped	OK Normal
ipnat	IP Network Address Translator					
Driver	c:\windows\system32\drivers\ipnat.sys	Kernel	No	Manual	Stopped	OK Normal
ipsec	IPSEC driver					

		c:\windows\system32\drivers\ipsec.sys	Kernel	Yes	System	Running	OK Normal
ipsraidn	ipsraidn	Kernel Driver	No	Yes			
	ipsraidn	Not Available	No	Disabled	Stopped	OK Normal	
irenum	IR Enumerator Service						
Driver	c:\windows\system32\drivers\irenum.sys	Kernel	No	Manual	Stopped	OK Normal	
isapnp	PnP ISA/EISA Bus Driver						
Driver	c:\windows\system32\drivers\isapnp.sys	Kernel	Yes	Boot	Running	OK Critical	
kbdclass	Keyboard Class Driver						
Driver	c:\windows\system32\drivers\kbdclass.sys	Kernel	Yes	System	Running	OK Normal	
ksecdd	KSecDD	c:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot	Running	
	ksecdd	Kernel Driver	Yes	Boot	Running	OK Normal	
lp6nds35	lp6nds35	Kernel Driver	No	Yes			
	lp6nds35	Not Available	No	Disabled	Stopped	OK Normal	
mnmd	mnmd	c:\windows\system32\drivers\mnmd.sys	Kernel Driver	Yes	System	Running	
	mnmd	Kernel Driver	Yes	System	Running	OK Normal	
modem	Modem	c:\windows\system32\drivers\modem.sys	Kernel Driver	No	Manual	Stopped	
	modem	Kernel Driver	No	Manual	Stopped	OK Normal	
mouclass	Mouse Class Driver						
Driver	c:\windows\system32\drivers\mouclass.sys	Kernel	Yes	System	Running	OK Normal	
mountmgr	Mount Point Manager						
Driver	c:\windows\system32\drivers\mountmgr.sys	Kernel	Yes	Boot	Running	OK Normal	
mraid35x	mraid35x	Kernel Driver	No	Yes			
	mraid35x	Not Available	No	Disabled	Stopped	OK Normal	
mrxdav	WebDav Client Redirector						
System Driver	c:\windows\system32\drivers\mrxdav.sys	File	No	Manual	Stopped	OK Normal	
mrxsmb	MRXSMB	c:\windows\system32\drivers\mrxsmb.sys	File System Driver	Yes	System	Running	
	mrxsmb	File System Driver	Yes	System	Running	OK Normal	
msfs	Msfs	c:\windows\system32\drivers\msfs.sys	File System Driver	Yes	System	Running	
	msfs	File System Driver	Yes	System	Running	OK Normal	
mssmbios	Microsoft System Management BIOS Driver						
Driver	c:\windows\system32\drivers\mssmbios.sys	Kernel	Yes	Manual	Running	OK Normal	

mup	Mup	c:\windows\system32\drivers\mup.sys	File System Driver	Yes	Boot	Running
	mup	File System Driver	Yes	Boot	Running	OK Normal
ndis	NDIS System Driver					
Driver	c:\windows\system32\drivers\ndis.sys	Kernel	Yes	Boot	Running	OK Normal
ndistapi	Remote Access NDIS TAPI Driver					
Driver	c:\windows\system32\drivers\ndistapi.sys	Kernel	Yes	Manual	Running	OK Normal
ndisuio	NDIS Usermode I/O Protocol					
Driver	c:\windows\system32\drivers\ndisuio.sys	Kernel	No	Manual	Stopped	OK Normal
ndiswan	Remote Access NDIS WAN Driver					
Driver	c:\windows\system32\drivers\ndiswan.sys	Kernel	Yes	Manual	Running	OK Normal
ndproxy	NDIS Proxy					
Driver	c:\windows\system32\drivers\ndproxy.sys	Kernel	Yes	Manual	Running	OK Normal
netbios	NetBIOS Interface					
System Driver	c:\windows\system32\drivers\netbios.sys	File	Yes	System	Running	OK Normal
netbt	NetBios over Tcpip					
Driver	c:\windows\system32\drivers\netbt.sys	Kernel	Yes	System	Running	OK Normal
nfrd960	nfrd960	Kernel Driver	No	Disabled	Stopped	OK Normal
npfs	Npfs	c:\windows\system32\drivers\npfs.sys	File System Driver	Yes	System	Running
	npfs	File System Driver	Yes	System	Running	OK Normal
ntfs	Ntfs	c:\windows\system32\drivers\ntfs.sys	File System Driver	Yes	Disabled	Running
	ntfs	File System Driver	Yes	Disabled	Running	OK Normal
null	Null	c:\windows\system32\drivers\null.sys	Kernel Driver	Yes	System	Running
	null	Kernel Driver	Yes	System	Running	OK Normal
parport	Parallel port driver					
Driver	c:\windows\system32\drivers\parport.sys	Kernel	Yes	Manual	Running	OK Normal
partmgr	Partition Manager					
Driver	c:\windows\system32\drivers\partmgr.sys	Kernel	Yes	Boot	Running	OK Normal
parvdm	Parvdm	c:\windows\system32\drivers\parvdm.sys	Kernel Driver	Yes	Auto	Running
	parvdm	Kernel Driver	Yes	Auto	Running	OK Normal

	No	Disabled	Stopped	OK	Normal
update	No	No			
Driver	Yes	Manual	Running	OK	Kernel
usbehci	Microsoft USB 2.0 Enhanced Host Controller				
Miniport Driver	Yes	Manual	Running	OK	Kernel
usbhub	USB2 Enabled Hub				
Driver	Yes	Manual	Running	OK	Kernel
usbstor	USB Mass Storage Driver				
Driver	No	Manual	Stopped	OK	Kernel
usbuhci	Microsoft USB Universal Host Controller				
Driver	Yes	Manual	Running	OK	Kernel
vga	VGA Display Controller.				
Kernel Driver	No	Manual	Stopped	OK	Kernel
vgasave	VGA Display Controller.				
Driver	Yes	System	Running	OK	Ignore
viaide	Vialde Not Available Kernel Driver				
Kernel Driver	No	Disabled	Stopped	OK	Normal
volsnap	Storage volumes				
Driver	Yes	Boot	Running	OK	Kernel
wanarp	Remote Access IP ARP Driver				
Driver	Yes	Manual	Running	OK	Kernel
wdica	WDICA Not Available Kernel Driver				
Kernel Driver	No	Manual	Stopped	OK	Ignore
wlbs	Network Load Balancing				
Driver	No	Manual	Stopped	OK	Kernel
	No	No			

[Signed Drivers]

Device Name	Signed	Device Class	Driver
Version	Driver Date	Manufacturer	INF
Name	Driver Name	Device ID	
Microsoft System Management BIOS DriverYes			

SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	
Not Available		ROOT\SYSTEM\0002	
Microcode Update Device	Yes	SYSTEM	
5.2.3790.0	10/1/2002	(Standard system devices)	
machine.inf		Not Available	
ROOT\SYSTEM\0001			
Plug and Play Software Device Enumerator	Yes	SYSTEM	
SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf		Not Available	
ROOT\SYSTEM\0000			
Terminal Server Mouse Driver	Yes	SYSTEM	
5.2.3790.0	10/1/2002	(Standard system devices)	
machine.inf		Not Available	
ROOT\RDP_MOU\0000			
Terminal Server Keyboard Driver	Yes	SYSTEM	
SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf		Not Available	
ROOT\RDP_KBD\0000			
Terminal Server Device Redirector	Yes	SYSTEM	
SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf		Not Available	
ROOT\RDPDR\0000			
Direct Parallel	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
ROOT\MS_PTMINIPORT\0000			
WAN Miniport (PPTP)	Yes	NET	
5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
Available	ROOT\MS_PPTMINIPORT\0000		Not Available
WAN Miniport (PPPOE)	Yes	NET	
5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
Available	ROOT\MS_PPPOEMINIPORT\0000		Not Available
WAN Miniport (IP)	Yes	NET	5.2.3790.0
10/1/2002	Microsoft	netrasa.inf	Not Available
Available	ROOT\MS_NDISWANIP\0000		Not Available
WAN Miniport (L2TP)	Yes	NET	
5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
Available	ROOT\MS_L2TPMINIPORT\0000		Not Available
Video Codecs	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	
Not Available		ROOT\MEDIA\MS_MMVID	
Legacy Video Capture Devices	Yes	MEDIA	
5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf		Not Available	
ROOT\MEDIA\MS_MMVCD			
Media Control Devices	Yes	MEDIA	
5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf		Not Available	
ROOT\MEDIA\MS_MMMCI			
Legacy Audio Drivers	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	
Not Available		ROOT\MEDIA\MS_MMDRV	

Audio Codecs	Yes	MEDIA	5.2.3790.0
10/1/2002	(Standard system devices)	wave.inf	
Not Available		ROOT\MEDIA\MS_MMACM	
Remote Access IP ARP Driver	Not Available		
LEGACYDRIVER	Not Available		Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_WANARP\0000		
volsnap	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
VGA Display Controller.	Not Available		
LEGACYDRIVER	Not Available		Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_VGASAVE\0000		
TDTCP	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
TCP/IP Protocol Driver	Not Available		
LEGACYDRIVER	Not Available		Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_TCPIP\0000		
RDPWD	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Remote Access Auto Connection Driver	Not Available		
LEGACYDRIVER	Not Available		Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_RASACD\0000		
Parvdm	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Partition Manager	Not Available	LEGACYDRIVER	
Not Available	Not Available		Not
Available	Not Available	Not Available	
Available	ROOT\LEGACY_PARTMGR\0000		
Null	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
NetBios over Tcpip	Not Available	LEGACYDRIVER	
Not Available	Not Available		Not
Available	Not Available	Not Available	
Available	Not Available	Not Available	
Available	Not Available	Not Available	
NDProxy	Not Available	LEGACYDRIVER	Not

Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_NDPROXY\0000		
NDIS Usermode I/O Protocol	Not Available		
	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_NDISUIO\0000		
Remote Access NDIS TAPI Driver	Not Available		
	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_NDISTAPI\0000		
NDIS System Driver	Not Available	LEGACYDRIVER	
	Not Available	Not Available	Not
Available	Not Available	Not Available	Not
	ROOT\LEGACY_NDIS\0000		
mountmgr	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_MOUNTMGR\0000		
mnmdd	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_MNMDD\0000		
ksecdd	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_KSECDD\0000		
IPSEC driver	Not Available	LEGACYDRIVER	
	Not Available	Not Available	Not
Available	Not Available	Not Available	
	ROOT\LEGACY_IPSEC\0000		
HTTP	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	ROOT\LEGACY_HTTP\0000	
Generic Packet Classifier	Not Available		
	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_GPC\0000		
Fips	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	ROOT\LEGACY_FIPS\0000	
dmload	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_DMLOAD\0000		
dmboot	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_DMBOOT\0000		
CRC Disk Filter Driver	Not Available		
	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not

Available	ROOT\LEGACY_CRCDISK\0000		
cpuz129	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available		
	ROOT\LEGACY_CPUZ129\0000		
Beep	Not Available	LEGACYDRIVER	Not
Available	Not Available	Not Available	Not
Available	Not Available	ROOT\LEGACY_BEEP\0000	
AFD Networking Support Environment	Not Available		
	LEGACYDRIVER	Not Available	Not
Available	Not Available	Not Available	Not
Available	ROOT\LEGACY_AFD\0000		
Generic volume	Yes	VOLUME 5.2.3790.0	
	10/1/2002 Microsoft	volume.inf	Not Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATU		
	REF0547C8COFFSET7E00LENGTH110F471A00		
Volume Manager	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ROOT\FTDISK\0000		
Logical Disk Manager	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ROOT\DMIO\0000		
ACPI Fixed Feature Button	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\FIXEDBUTTON\2&DABA3FF&0		
ACPI Power Button	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0C0C\3&61AAA01&0		
Intel(R) 631xESB/6321ESB/3100 Chipset SMBus Controller - 269B	Yes	SYSTEM 7.4.0.1005 5/15/2006 Intel	
	5000xzvp.inf	Not Available	
	PCI\VEN_8086&DEV_269B&SUBSYS_03321014&		
	REV_09\3&61AAA01&0&FB		
CD-ROM Drive	Yes	CDROM 5.2.3790.0	
	10/1/2002 (Standard CD-ROM drives)		
	cdrom.inf	Not Available	
	IDE\CDROMTSSTCORP_DVD-ROM_TS-		
	H352D_____LE01_____5&3B7DE513&0&0.0.0		
Primary IDE Channel	Yes	HDC 5.2.3790.0	
	10/1/2002 (Standard IDE ATA/ATAPI controllers)		
	mshdc.inf	Not Available	
	PCI\IDE\DECHANNEL\4&379AE00F&0&0		
Intel(R) 631xESB/6321ESB Ultra ATA Storage Controller - 269E	Yes	HDC 7.3.0.1010 11/18/2005	
	Intel	esb2ide.inf	Not Available
	PCI\VEN_8086&DEV_269E&SUBSYS_03321014&		
	REV_09\3&61AAA01&0&F9		

Communications Port	Yes	PORTS 5.2.3790.0	
	10/1/2002 (Standard port types)	msports.inf	
	Not Available	ACPI\PNP0501\2	
Printer Port Logical Interface	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	LPTENUM\MICROSOFTRAWPORT\6&231E718&		
0&LPT1			
ECP Printer Port	Yes	PORTS 5.2.3790.0	
	10/1/2002 (Standard port types)	msports.inf	
	Not Available	ACPI\PNP0401\4	
Communications Port	Yes	PORTS 5.2.3790.0	
	10/1/2002 (Standard port types)	msports.inf	
	Not Available	ACPI\PNP0501\1	
Generic Bus	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0A05\4&2AA4AD3D&0		
PS/2 Compatible Mouse	Yes	MOUSE	
	5.2.3790.0 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 timer	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0100\4&2AA4AD3D&0		
System speaker	Yes	SYSTEM 5.2.3790.0	
	10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0800\4&2AA4AD3D&0		
System CMOS/real time clock	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0B00\4&2AA4AD3D&0		
Programmable interrupt controller	Yes		
	SYSTEM 5.2.3790.0 10/1/2002 (Standard system		
devices)	machine.inf	Not Available	
	ACPI\PNP0000\4&2AA4AD3D&0		
Numeric data processor	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0C04\4&2AA4AD3D&0		
Direct memory access controller	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	
	ACPI\PNP0200\4&2AA4AD3D&0		
Motherboard resources	Yes	SYSTEM	
	5.2.3790.0 10/1/2002 (Standard system devices)		
	machine.inf	Not Available	

ACPI\PNP0C02\1F
 ISAPNP Read Data Port Yes SYSTEM
 5.2.3790.0 10/1/2002 (Standard system devices)
 machine.inf Not Available
 ISAPNP\READDATAPORT\0
 Intel(R) 631xESB/6321ESB/3100 Chipset LPC Interface
 Controller - 2670 Yes SYSTEM 7.4.0.1005
 5/15/2006 Intel 5000xzvp.inf Not
 Available
 PCI\VEN_8086&DEV_2670&SUBSYS_00000000&
 REV_09\3&61AAA01&0&F8
 Default Monitor Yes MONITOR
 5.1.2001.0 6/6/2001 (Standard monitor types)
 monitor.inf Not Available
 DISPLAY\DEFAULT_MONITOR\5&24B7E66&0&
 10000000&1C&04
 Default Monitor Yes MONITOR
 5.1.2001.0 6/6/2001 (Standard monitor types)
 monitor.inf Not Available
 DISPLAY\DEFAULT_MONITOR\5&24B7E66&0&
 10000001&1C&04
 Plug and Play Monitor Yes MONITOR
 5.1.2001.0 6/6/2001 (Standard monitor types)
 monitor.inf Not Available
 DISPLAY\IBM029A\5&24B7E66&0&10000082&1
 C&04
 ATI ES1000 Yes DISPLAY 8.19.4.0
 12/6/2005 ATI Technologies Inc. oem1.inf
 Not Available
 PCI\VEN_1002&DEV_515E&SUBSYS_03051014&
 REV_02\4&2014205D&0&20F0
 Intel(R) 82801 PCI Bridge - 244E Yes
 SYSTEM 7.0.0.1011 1/10/2005 Intel
 dmi_pci.inf Not Available
 PCI\VEN_8086&DEV_244E&SUBSYS_00000000&
 REV_D9\3&61AAA01&0&F0
 USB Root Hub Yes USB 5.2.3790.0
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB20\4&2FB71FA3&0
 Intel(R) 631xESB/6321ESB/3100 Chipset USB2 Enhanced Host
 Controller - 268C Yes USB 7.4.0.1005
 5/15/2006 Intel esb2usb.inf Not
 Available
 PCI\VEN_8086&DEV_268C&SUBSYS_03321014&
 REV_09\3&61AAA01&0&EF
 USB Root Hub Yes USB 5.2.3790.0
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&3648B3FB&0
 Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host
 Controller - 268B Yes USB 7.4.0.1005
 5/15/2006 Intel esb2usb.inf Not
 Available

PCI\VEN_8086&DEV_268B&SUBSYS_03321014&
 REV_09\3&61AAA01&0&EB
 USB Root Hub Yes USB 5.2.3790.0
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&13C311B0&0
 Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host
 Controller - 268A Yes USB 7.4.0.1005
 5/15/2006 Intel esb2usb.inf Not
 Available
 PCI\VEN_8086&DEV_268A&SUBSYS_03321014
 &REV_09\3&61AAA01&0&EA
 USB Root Hub Yes USB 5.2.3790.0
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&55CC539&0
 Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host
 Controller - 2689 Yes USB 7.4.0.1005
 5/15/2006 Intel esb2usb.inf Not
 Available
 PCI\VEN_8086&DEV_2689&SUBSYS_03321014&
 REV_09\3&61AAA01&0&E9
 USB Root Hub Yes USB 5.2.3790.0
 10/1/2002 (Standard USB Host Controller)
 usbport.inf Not Available
 USB\ROOT_HUB\4&3961E10E&0
 Intel(R) 631xESB/6321ESB/3100 Chipset USB Universal Host
 Controller - 2688 Yes USB 7.4.0.1005
 5/15/2006 Intel esb2usb.inf Not
 Available
 PCI\VEN_8086&DEV_2688&SUBSYS_03321014&
 REV_09\3&61AAA01&0&E8
 Broadcom NetXtreme Gigabit Ethernet No NET
 10.24.0.0 12/15/2006 Broadcom oem3.inf
 Not Available
 PCI\VEN_14E4&DEV_1659&SUBSYS_02C61014
 &REV_21\4&110C88BD&0&00E1
 Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root
 Port 2 - 2692 Yes SYSTEM 7.4.0.1005
 5/15/2006 Intel 5000xzvp.inf Not
 Available
 PCI\VEN_8086&DEV_2692&SUBSYS_00000000&
 REV_09\3&61AAA01&0&E1
 Broadcom NetXtreme Gigabit Ethernet No NET
 10.24.0.0 12/15/2006 Broadcom oem3.inf
 Not Available
 PCI\VEN_14E4&DEV_1659&SUBSYS_02C61014
 &REV_21\4&187919FE&0&00E0
 Intel(R) 631xESB/6321ESB/3100 Chipset PCI Express Root
 Port 1 - 2690 Yes SYSTEM 7.4.0.1005
 5/15/2006 Intel 5000xzvp.inf Not
 Available
 PCI\VEN_8086&DEV_2690&SUBSYS_00000000&
 REV_09\3&61AAA01&0&E0

Intel(R) 5000 Series Chipset FBD Registers - 25F6 Yes
 SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F6&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&B0
 Intel(R) 5000 Series Chipset FBD Registers - 25F5 Yes
 SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F5&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&A8
 Intel(R) 5000 Series Chipset Reserved Registers - 25F3 Yes
 SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F3&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&98
 Intel(R) 5000 Series Chipset Reserved Registers - 25F1 Yes
 SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F1&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&88
 Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&82
 Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&81
 Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0
 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&80
 Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7
 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25E7&SUBSYS_00000000&
 REV_B1\3&61AAA01&0&38
 Intel(R) PRO/1000 PT Dual Port Server Adapter Yes
 NET 9.6.31.0 10/31/2006 Intel
 oem5.inf Not Available
 PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
 &REV_06\4&30A54032&0&0130
 Intel(R) PRO/1000 PT Dual Port Server Adapter Yes
 NET 9.6.31.0 10/31/2006 Intel
 oem5.inf Not Available
 PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
 &REV_06\4&30A54032&0&0030
 Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9
 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
 5000xzvp.inf Not Available
 PCI\VEN_8086&DEV_25F9&SUBSYS_00000000&


```

REV_B1\3&61AAA01&0&30
Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5
Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25E5&SUBSYS_00000000&
REV_B1\3&61AAA01&0&28
Intel(R) PRO/1000 PT Dual Port Server Adapter Yes
NET 9.6.31.0 10/31/2006 Intel
oem5.inf Not Available
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_06\4&8E1D94C&0&0120
Intel(R) PRO/1000 PT Dual Port Server Adapter Yes
NET 9.6.31.0 10/31/2006 Intel
oem5.inf Not Available
PCI\VEN_8086&DEV_105E&SUBSYS_125E8086
&REV_06\4&8E1D94C&0&0020
Intel(R) 5000 Series Chipset PCI Express x8 Port 4-5 - 25F8
Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25F8&SUBSYS_00000000&
REV_B1\3&61AAA01&0&20
Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3
Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25E3&SUBSYS_00000000&
REV_B1\3&61AAA01&0&18
Intel(R) 6311ESB/6321ESB PCI Express to PCI-X Bridge -
350C Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_350C&SUBSYS_00000000&
REV_01\4&1EE18D9A&0&0310
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E2 -
3514 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_3514&SUBSYS_00000000&
REV_01\5&1AA5474&0&080010
Disk drive Yes DISKDRIVE 5.2.3790.0
10/1/2002 (Standard disk drives) disk.inf Not
Available
SCSI\DISK&VEN_ADAPTEC&PROD_ARRAY&R
EV_V1.0\7&1B84E868&0&400
IBM ServeRAID 8k/8k-1 Controller Yes
SCSIADAPTER 5.1.0.9206 4/13/2006 Adaptec
oem0.inf Not Available
PCI\VEN_9005&DEV_0286&SUBSYS_95801014&
REV_02\6&305972A8&0&00000010
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 -
3510 Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_3510&SUBSYS_00000000&
REV_01\5&1AA5474&0&000010
Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available

```

```

PCI\VEN_8086&DEV_3500&SUBSYS_00000000&
REV_01\4&1EE18D9A&0&0010
Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
Yes SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25F7&SUBSYS_00000000&
REV_B1\3&61AAA01&0&10
Intel(R) 5000P Chipset Memory Controller Hub - 25D8 Yes
SYSTEM 7.4.0.1005 5/15/2006 Intel
5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25D8&SUBSYS_00000000
&REV_B1\3&61AAA01&0&00
PCI bus Yes SYSTEM 5.2.3790.0 10/1/2002
(Standard system devices) machine.inf
Not Available
ACPI\PNP0A03\2&DABA3FF&0
Intel Processor Yes PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_X86_FAMILY_6_MODEL_23\3
Intel Processor Yes PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_X86_FAMILY_6_MODEL_23\2
Intel Processor Yes PROCESSOR
5.2.3790.1830 10/1/2002 Intel cpu.inf
Not Available ACPI\GENUINEINTEL_-
_X86_FAMILY_6_MODEL_23\1
Intel Processor Yes PROCESSOR
5.2.3790.1830 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
5.2.3790.0 10/1/2002 (Standard computers) hal.inf
Not Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available
Not Available Not Available Not Available
Available Not Available Not Available
HTREE\ROOT\0
[Environment Variables]
Variable Value User Name
ClusterLogC:\WINDOWS\Cluster\cluster.log
<SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe
<SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>

```

```

NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%System
Root%\System32\Wbem;c:\tools;c:\tools\util;C:\Program
Files\Microsoft SQL Server\80\Tools\Binn\C:\Program
Files\Microsoft SQL Server\90\Tools\bin\C:\Program
Files\Microsoft SQL Server\90\DTS\Binn\C:\Program
Files\Microsoft SQL
Server\90\Tools\Binn\VSShell\Comman7\IDE\C:\Program
Files\Intel\DMIX <SYSTEM>
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF
;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE x86
<SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 6 Model 23
Stepping 6, GenuineIntel <SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 1706 <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
windir %SystemRoot% <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT
AUTHORITY\LOCAL SERVICE
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
FCLIENT38\Administrator
TMP %USERPROFILE%\Local Settings\Temp
FCLIENT38\Administrator
[Print Jobs]
Document Size Owner Notify Status Time
Submitted Start Time Until TimeElapsed Time Pages
Printed Job ID Priority ParametersDriver Print
Processor Host Print Queue Data Type Name
[Network Connections]

```

Local Name	Remote Name	Type	Status
x:	\\fsserv\rayDisk	Persistent Connection	
y:	\\fsserv\edrive	Disk Persistent Connection	

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start Time	Version	Size
system idle process		Not Available	0	0				
Available	Not Available	Not Available	Not Available	Not Available	Not Available			
system	Not Available	4	8	0				
smss.exe	Not Available	364	11	204800				
csrss.exe	Not Available	852	13	Not Available				
winlogon.exe	c:\windows\system32\winlogon.exe	960	13	204800	1413120	8/28/2008 2:39 PM	5.2.3790.1830	497.00 KB (508,928 bytes)
services.exe	c:\windows\system32\services.exe	1004	9	204800	1413120	8/28/2008 2:39 PM	5.2.3790.1830	107.50 KB (110,080 bytes)
lsass.exe	c:\windows\system32\lsass.exe	204800	1413120	8/28/2008 2:39 PM	5.2.3790.0	(srv03_rtm.030324-2048)		13.00 KB (13,312 bytes)
svchost.exe	c:\windows\system32\svchost.exe	1216	8	204800	1413120	8/28/2008 2:39 PM	5.2.3790.1830	14.00 KB (14,336 bytes)
svchost.exe	Not Available	1328	8					
svchost.exe	Not Available	1384	8					
svchost.exe	Not Available	1428	8					

8/28/2008 2:39 PM	Not Available	Not Available	Not Available
svchost.exe	c:\windows\system32\svchost.exe	1476	8
8/28/2008 2:39 PM	5.2.3790.1830	14.00 KB (14,336 bytes)	
msdtc.exe	Not Available	396	8
inetinfo.exe	c:\windows\system32\inetinfo.exe	604	8
svchost.exe	Not Available	1368	8
svchost.exe	c:\windows\system32\svchost.exe	1524	8
explorer.exe	c:\windows\explorer.exe	2008	8
svchost.exe	c:\windows\system32\svchost.exe	744	8
bmmlisten.exe	c:\slave\bmmlisten.exe	872	8
wmiprvse.exe	Not Available	1088	8
psaxesvc.exe	c:\windows\psaxesvc.exe	692	8
cmd.exe	c:\windows\system32\cmd.exe	1640	8
msinfo32.exe	c:\program files\common files\microsoft shared\msinfo\msinfo32.exe	920	8

(srv03_sp1_rtm.050324-1447)	42.00 KB (43,008 bytes)		
wmiprvse.exe	Not Available	1288	8
Available	Not Available	Not Available	Not Available

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer
winlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	497.00 KB (508,928 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
ntdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	748.50 KB (766,464 bytes)	3/25/2003 8:00 AM	Microsoft Corporation
kernel32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,014.00 KB (1,038,336 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
advapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	605.50 KB (620,032 bytes)	3/25/2003 8:00 AM	Microsoft Corporation
rpcrt4	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	627.00 KB (642,048 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
crypt32	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	582.00 KB (595,968 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
msasn1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	56.50 KB (57,856 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
msvcrt	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)	340.50 KB (348,672 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
user32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	574.50 KB (588,288 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
gdi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	273.00 KB (279,552 bytes)	8/7/2006 1:38 PM	Microsoft Corporation
nddeapi	5.2.3790.0 (srv03_rtm.030324-2048)	16.00 KB (16,384 bytes)	3/25/2003 8:00 AM	Microsoft Corporation

profmap 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 22.50 KB (23,040 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\profmap.dll
 netapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 341.50 KB (349,696 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\netapi32.dll
 userenv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 771.00 KB (789,504 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\userenv.dll
 psapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 20.00 KB (20,480 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\psapi.dll
 regapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 55.00 KB (56,320 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\regapi.dll
 secur32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 64.00 KB (65,536 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\secur32.dll
 setupapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 1.03 MB (1,079,808 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\setupapi.dll
 version 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 18.00 KB (18,432 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\version.dll
 winsta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 54.50 KB (55,808 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\winsta.dll
 ws2_32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 82.00 KB (83,968 bytes) 8/7/2006 1:37 PM
 Microsoft Corporation
 c:\windows\system32\ws2_32.dll
 ws2help 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 19.50 KB (19,968 bytes) 8/7/2006 1:37 PM
 Microsoft Corporation
 c:\windows\system32\ws2help.dll
 msgina 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 1.16 MB (1,211,904 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\msgina.dll
 shsvcs 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 131.50 KB (134,656 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\shsvcs.dll
 shlwapi 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 313.50 KB (321,024 bytes) 8/7/2006 1:38 PM

Microsoft Corporation
 c:\windows\system32\shlwapi.dll
 sfc 5.2.3790.0 (srv03_rtm.030324-2048) 4.50 KB
 (4,608 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\sfc.dll
 sfc_os 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 138.00 KB (141,312 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\sfc_os.dll
 wintrust 5.131.3790.1830 (srv03_sp1_rtm.050324-1447)
 162.00 KB (165,888 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\wintrust.dll
 imagehlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 145.50 KB (148,992 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\imagehlp.dll
 ole32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 1.19 MB (1,245,184 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\ole32.dll
 comctl32 6.0 (srv03_sp1_rtm.050324-1447) 1.00 MB
 (1,051,136 bytes) 3/24/2005 9:41 PM Microsoft
 Corporation c:\windows\winsxs\x86_microsoft.windows.common-
 controls_6595b64144ccf1df_6.0.3790.1830_x-
 ww_7ae38ccf\comctl32.dll
 winscard 5.2.3790.0 (srv03_rtm.030324-2048) 98.50
 KB (100,864 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\winscard.dll
 wtsapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 19.00 KB (19,456 bytes) 8/7/2006 1:37 PM
 Microsoft Corporation
 c:\windows\system32\wtsapi32.dll
 sxs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 743.50 KB (761,344 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\sxs.dll
 winmm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 172.50 KB (176,640 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\winmm.dll
 shell32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 7.99 MB (8,379,392 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\shell32.dll
 wldap32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 174.50 KB (178,688 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\wldap32.dll
 rsaenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 183.98 KB (188,392 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation

c:\windows\system32\rsaenh.dll
 ati2evxx 6.14.10.4123 46.50 KB (47,616 bytes)
 12/6/2005 9:38 PM ATI Technologies Inc.
 c:\windows\system32\ati2evxx.dll
 csd.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 100.00 KB (102,400 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\csd.dll
 dimsntfy 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 19.00 KB (19,456 bytes) 8/7/2006 1:42 PM
 Microsoft Corporation
 c:\windows\system32\dimsntfy.dll
 wlnotify 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 94.50 KB (96,768 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\wlnotify.dll
 mpr 5.2.3790.0 (srv03_rtm.030324-2048) 56.00
 KB (57,344 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\mpr.dll
 oleaut32 5.2.3790.1830 543.00 KB (556,032 bytes)
 3/25/2003 8:00 AM Microsoft Corporation
 c:\windows\system32\oleaut32.dll
 winspool 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 147.00 KB (150,528 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\winspool.drv
 comctl32 5.82 (srv03_sp1_rtm.050324-1447) 585.00
 KB (599,040 bytes) 3/24/2005 9:41 PM Microsoft
 Corporation c:\windows\winsxs\x86_microsoft.windows.common-
 controls_6595b64144ccf1df_5.82.3790.1830_x-
 ww_1b6f474a\comctl32.dll
 uxtheme 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 202.00 KB (206,848 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\uxtheme.dll
 samlib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 46.50 KB (47,616 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\samlib.dll
 cscur 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 319.50 KB (327,168 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\cscur.dll
 drprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 14.00 KB (14,336 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\drprov.dll
 ntlanman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 43.50 KB (44,544 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\ntlanman.dll
 netui0 5.2.3790.0 (srv03_rtm.030324-2048) 75.50
 KB (77,312 bytes) 3/25/2003 8:00 AM Microsoft

Corporation c:\windows\system32\netui0.dll
netui1 5.2.3790.0 (srv03_rtm.030324-2048) 184.00
KB (188,416 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\netui1.dll
davclnt 5.2.3790.0 (srv03_rtm.030324-2048) 23.50
KB (24,064 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\davclnt.dll
mprui 5.2.3790.0 (srv03_rtm.030324-2048) 49.00
KB (50,176 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\mprui.dll
netui2 5.2.3790.0 (srv03_rtm.030324-2048) 309.50
KB (316,928 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\netui2.dll
comdlg32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
274.50 KB (281,088 bytes) 3/25/2003 8:00 AM
Microsoft Corporation
c:\windows\system32\comdlg32.dll
netmsg 5.2.3790.0 (srv03_rtm.030324-2048) 178.00
KB (182,272 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\netmsg.dll
clbcatq 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)
502.50 KB (514,560 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\clbcatq.dll
comres 2001.12.4720.0 (srv03_rtm.030324-2048) 778.00
KB (796,672 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\comres.dll
ntmarta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
120.50 KB (123,392 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ntmarta.dll
xpsp2res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
2.76 MB (2,897,920 bytes) 8/7/2006 1:41 PM
Microsoft Corporation
c:\windows\system32\xpsp2res.dll
services 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
107.50 KB (110,080 bytes) 3/25/2003 8:00 AM
Microsoft Corporation
c:\windows\system32\services.exe
ncobjapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
36.00 KB (36,864 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ncobjapi.dll
msvcp60 6.05.2144.0 388.00 KB (397,312 bytes)
3/25/2003 8:00 AM Microsoft Corporation
c:\windows\system32\msvcp60.dll
scsvr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
327.00 KB (334,848 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\scsvr.dll
authz 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
66.50 KB (68,096 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\authz.dll

umpnpgmgr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
126.50 KB (129,536 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\umpnpgmgr.dll
eventlog 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
67.50 KB (69,120 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\eventlog.dll
lsass 5.2.3790.0 (srv03_rtm.030324-2048) 13.00
KB (13,312 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\lsass.exe
lsasrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
803.00 KB (822,272 bytes) 3/25/2003 8:00 AM
Microsoft Corporation
c:\windows\system32\lsasrv.dll
ntdsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
71.00 KB (72,704 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ntdsapi.dll
dnsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
153.50 KB (157,184 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\dnsapi.dll
samsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
450.50 KB (461,312 bytes) 3/25/2003 8:00 AM
Microsoft Corporation
c:\windows\system32\samsrv.dll
cryptdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
32.00 KB (32,768 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\cryptdll.dll
msprivs 5.2.3790.0 (srv03_rtm.030324-2048) 46.50
KB (47,616 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\msprivs.dll
kerberos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
340.50 KB (348,672 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\kerberos.dll
msv1_0 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
141.00 KB (144,384 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\msv1_0.dll
iphlpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
92.50 KB (94,720 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\iphlpapi.dll
netlogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
409.50 KB (419,328 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\netlogon.dll
w32time 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
222.00 KB (227,328 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\w32time.dll

schannel 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
141.00 KB (144,384 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\schannel.dll
wdigest 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
74.00 KB (75,776 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\wdigest.dll
rassfm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
23.00 KB (23,552 bytes) 8/7/2006 1:39 PM
Microsoft Corporation
c:\windows\system32\rassfm.dll
kdcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
213.50 KB (218,624 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
1.45 MB (1,516,032 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ntdsa.dll
esent 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
1,022.50 KB (1,047,040 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\esent.dll
ntdsatq 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
29.50 KB (30,208 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ntdsatq.dll
mswsock 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
250.50 KB (256,512 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\mswsock.dll
scecli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
186.50 KB (190,976 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\scecli.dll
ws03res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
793.50 KB (812,544 bytes) 8/7/2006 1:42 PM
Microsoft Corporation
c:\windows\system32\ws03res.dll
hnetcfg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
343.50 KB (351,744 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\hnetcfg.dll
wshtcpip 5.2.3790.0 (srv03_rtm.030324-2048) 18.00
KB (18,432 bytes) 3/25/2003 8:00 AM Microsoft
Corporation c:\windows\system32\wshtcpip.dll
ipsecsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
180.50 KB (184,832 bytes) 8/7/2006 1:38 PM
Microsoft Corporation
c:\windows\system32\ipsecsvc.dll
oakley 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
264.00 KB (270,336 bytes) 8/7/2006 1:38 PM

winipsec	Microsoft Corporation c:\windows\system32\oakley.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 35.50 KB (36,352 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\winipsec.dll	wiarpc	130.00 KB (133,120 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\wkssvc.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 32.50 KB (33,280 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\wiarpc.dll	trkwks	c:\windows\system32\sens.dll 5.2.3790.0 (srv03_rtm.030324-2048) 85.00 KB (87,040 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\trkwks.dll
pstorsvc	5.2.3790.0 (srv03_rtm.030324-2048) 24.00 KB (24,576 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\pstorsvc.dll	aelupsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 26.00 KB (26,624 bytes) 8/7/2006 1:42 PM Microsoft Corporation c:\windows\system32\aelupsvc.dll	comsvcs	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 1.19 MB (1,248,256 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\wbem\wmisvc.dll
psbase	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 84.00 KB (86,016 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\psbase.dll	apphelp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 146.50 KB (150,016 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\apphelp.dll	browser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 76.50 KB (78,336 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\browser.dll
dssenh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 139.98 KB (143,336 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\dssenh.dll	cryptsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 55.50 KB (56,832 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\cryptsvc.dll	netman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 258.50 KB (264,704 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\netman.dll
wlbsctrl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 82.00 KB (83,968 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\wlbsctrl.dll	certcli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 227.00 KB (232,448 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\certcli.dll	mprapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 89.00 KB (91,136 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\mprapi.dll
w3ssl	6.0.3790.0 (srv03_rtm.030324-2048) 15.00 KB (15,360 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\w3ssl.dll	atl	3.05.2283 83.00 KB (84,992 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\atl.dll	activeds	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 194.00 KB (198,656 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\activeds.dll
strmfilt	6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 84.00 KB (86,016 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\strmfilt.dll	vssapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 548.00 KB (561,152 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\vssapi.dll	adslsdp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 146.00 KB (149,504 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\adslsdp.dll
httpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\httpapi.dll	dmserver	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 25.50 KB (26,112 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\dmserver.dll	credui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 162.00 KB (165,888 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\credui.dll
svchost	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\svchost.exe	es	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 233.00 KB (238,592 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\es.dll	rtutils	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 34.50 KB (35,328 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\rtutils.dll
rpcss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 406.00 KB (415,744 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\rpcss.dll	srvsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\srvsvc.dll	netshell	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.73 MB (1,812,992 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\netshell.dll
schedsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 197.50 KB (202,240 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\schedsvc.dll	pchsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 39.00 KB (39,936 bytes) 8/7/2006 1:39 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchsvc.dll	clusapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 60.00 KB (61,440 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\clusapi.dll
msidle	6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 6.50 KB (6,656 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\msidle.dll	seclogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 18.50 KB (18,944 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\seclogon.dll	rasapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 239.50 KB (245,248 bytes) 3/25/2003 8:00 AM Microsoft Corporation c:\windows\system32\rasapi32.dll
audiosrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 40.50 KB (41,472 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\audiosrv.dll	sens	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.50 KB (37,376 bytes) 8/7/2006 1:38 PM Microsoft Corporation	rasman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 61.50 KB (62,976 bytes) 3/25/2003 8:00 AM
wkssvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)				

tapi32	Microsoft Corporation c:\windows\system32\rasman.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 179.50 KB (183,808 bytes) 8/7/2006 1:38 PM	wbemess	404.00 KB (413,696 bytes) 8/7/2006 1:38 PM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 271.50 KB (278,016 bytes) 8/7/2006 1:39 PM	admwprox	62.50 KB (64,000 bytes) 8/7/2006 1:39 PM Microsoft Corporation c:\windows\system32\inetsrv\coadmin.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes) 8/7/2006 1:38 PM
wininet	Microsoft Corporation c:\windows\system32\tapi32.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 646.00 KB (661,504 bytes) 8/7/2006 1:38 PM	rasdlg	Microsoft Corporation c:\windows\system32\wbem\wbemess.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 663.00 KB (678,912 bytes) 3/25/2003 8:00 AM	iiscfg	Microsoft Corporation c:\windows\system32\admwprox.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 1.08 MB (1,133,056 bytes) 8/7/2006 1:39 PM
wzcsapi	Microsoft Corporation c:\windows\system32\wininet.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 41.00 KB (41,984 bytes) 8/7/2006 1:37 PM	netrap	Microsoft Corporation c:\windows\system32\rasdlg.dll 5.2.3790.0 (srv03_rtm.030324-2048) 11.50 KB (11,776 bytes) 3/25/2003 8:00 AM	metadata	Microsoft Corporation c:\windows\system32\inetsrv\iiscfg.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 229.00 KB (234,496 bytes) 8/7/2006 1:39 PM
wzcsvc	Microsoft Corporation c:\windows\system32\wzcsapi.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 364.50 KB (373,248 bytes) 8/7/2006 1:37 PM	rasadhlp	Microsoft Corporation c:\windows\system32\netrap.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 7.50 KB (7,680 bytes) 8/7/2006 1:38 PM	msxml3	Microsoft Corporation c:\windows\system32\inetsrv\metadata.dll 8.70.1104.0 1.06 MB (1,107,456 bytes) 8/7/2006 1:38 PM
wmi	Microsoft Corporation c:\windows\system32\wzcsvc.dll 5.2.3790.0 (srv03_rtm.030324-2048) 6.50 KB (6,656 bytes) 3/25/2003 8:00 AM	ncprov	Microsoft Corporation c:\windows\system32\rasadhlp.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 46.50 KB (47,616 bytes) 8/7/2006 1:39 PM	svcxext	Microsoft Corporation c:\windows\system32\msxml3.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 43.50 KB (44,544 bytes) 8/7/2006 1:39 PM
dhcpcsvc	Microsoft Corporation c:\windows\system32\wmi.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 113.50 KB (116,224 bytes) 3/25/2003 8:00 AM	netcfgx	Microsoft Corporation c:\windows\system32\ncprov.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 763.00 KB (781,312 bytes) 8/7/2006 1:38 PM	security	Microsoft Corporation c:\windows\system32\inetsrv\svcxext.dll 5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes) 3/25/2003 8:00 AM
wbemcomn	Microsoft Corporation c:\windows\system32\dhcpcsvc.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 221.00 KB (226,304 bytes) 8/7/2006 1:39 PM	ntlsapi	Microsoft Corporation c:\windows\system32\netcfgx.dll 5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB (8,192 bytes) 3/25/2003 8:00 AM	iismap	Microsoft Corporation c:\windows\system32\security.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 58.50 KB (59,904 bytes) 8/7/2006 1:38 PM
wbemcore	Microsoft Corporation c:\windows\system32\wbem\wbemcomn.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50 KB (509,440 bytes) 8/7/2006 1:39 PM	wbemcons	Microsoft Corporation c:\windows\system32\ntlsapi.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 45.50 KB (46,592 bytes) 8/7/2006 1:39 PM	wamreg	Microsoft Corporation c:\windows\system32\iismap.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 54.50 KB (55,808 bytes) 8/7/2006 1:39 PM
esscli	Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 250.00 KB (256,000 bytes) 8/7/2006 1:39 PM	inetinfo	Microsoft Corporation c:\windows\system32\wbem\wbemcons.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 8/7/2006 1:39 PM	iisw3adm	Microsoft Corporation c:\windows\system32\inetsrv\wamreg.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 211.00 KB (216,064 bytes) 8/7/2006 1:39 PM
fastprox	Microsoft Corporation c:\windows\system32\wbem\esscli.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 471.00 KB (482,304 bytes) 8/7/2006 1:39 PM	iisutil	Microsoft Corporation c:\windows\system32\inetinfo.exe 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 164.00 KB (167,936 bytes) 8/7/2006 1:39 PM	w3cache	Microsoft Corporation c:\windows\system32\inetsrv\iisw3adm.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 19.00 KB (19,456 bytes) 8/7/2006 1:39 PM
wbemsvc	Microsoft Corporation c:\windows\system32\wbem\fastprox.dll 5.2.3790.0 (srv03_rtm.030324-2048) 42.50 KB (43,520 bytes) 8/4/2006 6:53 PM	rpcref	Microsoft Corporation c:\windows\system32\inetsrv\iisutil.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 4.00 KB (4,096 bytes) 8/7/2006 1:39 PM	w3tp	Microsoft Corporation c:\windows\system32\inetsrv\w3cache.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 13.00 KB (13,312 bytes) 8/7/2006 1:39 PM
wmiutils	Microsoft Corporation c:\windows\system32\wbem\wbemsvc.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 8/7/2006 1:38 PM	iisrtl	Microsoft Corporation c:\windows\system32\inetsrv\rpcref.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 138.50 KB (141,824 bytes) 8/7/2006 1:38 PM	lonsint	Microsoft Corporation c:\windows\system32\inetsrv\w3tp.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 13.00 KB (13,312 bytes) 8/7/2006 1:39 PM
repdrvfs	Microsoft Corporation c:\windows\system32\wbem\wmiutils.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 172.50 KB (176,640 bytes) 8/7/2006 1:39 PM	iisadmin	Microsoft Corporation c:\windows\system32\iisrtl.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 21.00 KB (21,504 bytes) 8/7/2006 1:39 PM	explorer	Microsoft Corporation c:\windows\system32\inetsrv\lonsint.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 1.00 MB (1,050,624 bytes) 8/7/2006 1:38 PM
wmiprvsd	Microsoft Corporation c:\windows\system32\repdrvfs.dll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	coadmin	Microsoft Corporation c:\windows\system32\iisadmin.dll 6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	browseui	Microsoft Corporation c:\windows\explorer.exe 6.0.3790.1830 (srv03_sp1_rtm.050324-1447)

1,009.00 KB (1,033,216 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\browseui.dll
 shdocvw 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 1.43 MB (1,502,720 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\shdocvw.dll
 cryptui 5.131.3790.1830 (srv03_sp1_rtm.050324-1447)
 496.50 KB (508,416 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\cryptui.dll
 themeui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 377.50 KB (386,560 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\themeui.dll
 msimg32 5.2.3790.0 (srv03_rtm.030324-2048) 4.50 KB
 (4,608 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\msimg32.dll
 actxprxy 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 96.50 KB (98,816 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\actxprxy.dll
 linkinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 19.00 KB (19,456 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\linkinfo.dll
 ntshru1 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 140.00 KB (143,360 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\ntshru1.dll
 urlmon 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 673.00 KB (689,152 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\urlmon.dll
 webcheck 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 272.50 KB (279,040 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\webcheck.dll
 wsock32 5.2.3790.0 (srv03_rtm.030324-2048) 22.00
 KB (22,528 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\wsock32.dll
 stobject 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 120.50 KB (123,392 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\stobject.dll
 batmeter 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 31.50 KB (32,256 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\batmeter.dll
 powrprof 6.00.3790.1830 (srv03_sp1_rtm.050324-1447)
 16.50 KB (16,896 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation

c:\windows\system32\powrprof.dll
 termsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 239.00 KB (244,736 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\termsrv.dll
 icaapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 12.50 KB (12,800 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\icaapi.dll
 mstlsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 116.00 KB (118,784 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\mstlsapi.dll
 rdpwsx 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 101.63 KB (104,072 bytes) 8/7/2006 1:38 PM
 Microsoft Corporation
 c:\windows\system32\rdpwsx.dll
 bmlisten 2, 1, 0, 2 156.00 KB (159,744 bytes) 8/7/2006
 1:31 PM IBM xSeries Server Performance
 c:\slave\bmlisten.exe
 psaxesvc 1.93 97.26 KB (99,592 bytes)
 8/28/2008 4:01 PM Sysinternals
 c:\windows\psaxesvc.exe
 cmd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 379.00 KB (388,096 bytes) 3/25/2003 8:00 AM
 Microsoft Corporation
 c:\windows\system32\cmd.exe
 msinfo32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 42.00 KB (43,008 bytes) 8/7/2006 1:39 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)
 8/7/2006 1:38 PM Microsoft Corporation
 c:\windows\system32\mfc42u.dll
 odbc32 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 240.00
 KB (245,760 bytes) 8/7/2006 1:38 PM Microsoft
 Corporation c:\windows\system32\odbc32.dll
 odbcint 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 92.00
 KB (94,208 bytes) 8/7/2006 1:38 PM Microsoft
 Corporation c:\windows\system32\odbcint.dll
 msinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 376.00 KB (385,024 bytes) 8/7/2006 1:39 PM
 Microsoft Corporation
 c:\windows\pchealth\helpctr\binaries\msinfo.dll
 riched32 5.2.3790.0 (srv03_rtm.030324-2048) 3.50 KB
 (3,584 bytes) 3/25/2003 8:00 AM Microsoft
 Corporation c:\windows\system32\riched32.dll
 riched20 5.31.23.1224 439.00 KB (449,536 bytes)
 8/7/2006 1:38 PM Microsoft Corporation
 c:\windows\system32\riched20.dll
 wbemprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)

20.50 KB (20,992 bytes) 8/7/2006 1:39 PM
 Microsoft Corporation
 c:\windows\system32\wbem\wbemprox.dll

[Services]

Display Name	Name	State	Start Mode	Service
Type	Path	Error Control	Start Name	Tag ID
Application Experience Lookup Service	AeLookupSvc	Running	Auto	Share Process
	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
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\v2.0.50727\aspn	Normal	NT	AUTHORITY\NetworkService
Ati HotKey Poller	Ati HotKey Poller	Stopped	Manual	Own Process
	c:\windows\system32\ati2evxx.exe	Normal	LocalSystem	0
Windows Audio Process	AudioSrv	Running	Auto	Share
	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 Process	Browser	Running	Auto	Share
	c:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0
Indexing Service Process	CiSvc	Stopped	Disabled	Share
	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\v2.0.50727\msc			

```

orsvw.exe Ignore LocalSystem 0
COM+ System Application COMSysApp Stopped
Manual Own Process
c:\windows\system32\dlhhost.exe
/processid:{02d4b3f1-fd88-11d1-960d-00805fc79235} Normal
LocalSystem 0
Cryptographic Services CryptSvc Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
DCOM Server Process Launcher DcomLaunch Running
Auto Share Process
c:\windows\system32\svchost.exe -k dcomlaunch
Normal LocalSystem 0
Distributed File System Dfs Stopped Manual
Own Process
c:\windows\system32\dfssvc.exe Normal
LocalSystem 0
DHCP Client Dhcp Stopped Manual Share
Process c:\windows\system32\svchost.exe -k networkservice
Normal NT AUTHORITY\NetworkService
0
Logical Disk Manager Administrative Service
dmadmin Stopped Manual Share Process
c:\windows\system32\dmadmin.exe /com Normal
LocalSystem 0
Logical Disk Managerdmsrver 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 Stopped Manual
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 SSLHTTPFilter Running Manual Share
Process c:\windows\system32\lsass.exe Normal
LocalSystem 0

```

```

IIS Admin Service IISADMIN Running Auto
Share Process
c:\windows\system32\inetrv\inetinfo.exe Normal
LocalSystem 0
IMAPI CD-Burning COM Service ImapiService
Stopped Disabled Own Process
c:\windows\system32\imapi.exe Normal
LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own
Process c:\windows\system32\ismserv.exe Normal
LocalSystem 0
Kerberos Key Distribution Center kdc Stopped
Disabled Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Server lanmanserver Running Auto Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Workstation lanmanworkstation Running Auto
Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
License Logging LicenseService Stopped Disabled
Own Process c:\windows\system32\llsrv.exe
Normal NT AUTHORITY\NetworkService
0
TCP/IP NetBIOS Helper LmHosts Running Auto
Share Process
c:\windows\system32\svchost.exe -k localservice
Normal NT AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped
Disabled Own Process
c:\windows\system32\mnmsrvc.exe Normal
LocalSystem 0
Distributed Transaction Coordinator MSDTC Running
Auto Own Process
c:\windows\system32\msdtc.exe Normal NT
AUTHORITY\NetworkService 0
Windows Installer MSIServer Stopped Manual Share
Process c:\windows\system32\msiexec.exe /v Normal
LocalSystem 0
Network DDE NetDDE Stopped Disabled Share
Process c:\windows\system32\netdde.exe Normal
LocalSystem 0
Network DDE DSDMNetDDEdsdm 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 NtFrs Stopped Manual Own
Process c:\windows\system32\ntfrs.exe Ignore
LocalSystem 0
NT LM Security Support Provider NtLmSsp Stopped
Manual Share Process
c:\windows\system32\lsass.exe Normal
LocalSystem 0
Removable Storage NtmsSvc Stopped Manual Share
Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
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 Running Auto
Share Process c:\windows\system32\lsass.exe
Normal LocalSystem 0
Protected Storage ProtectedStorage Running Auto
Share Process c:\windows\system32\lsass.exe
Normal LocalSystem 0
Remote Access Auto Connection Manager RasAuto Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Access Connection Manager RasMan Stopped
Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Desktop Help Session Manager RDSessMgr
Stopped Manual Own Process
c:\windows\system32\sessmgr.exe Normal
LocalSystem 0
Routing and Remote Access RemoteAccess Stopped
Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Remote Registry RemoteRegistry Running Auto
Share Process
c:\windows\system32\svchost.exe -k regsvc Normal
NT AUTHORITY\LocalService 0
Remote Procedure Call (RPC) Locator RpcLocator
Stopped Manual Own Process
c:\windows\system32\locator.exe Normal NT

```


AUTHORITY\NetworkService	0				
Remote Procedure Call (RPC) Share Process	RpcSs	Running	Auto		
	c:\windows\system32\svchost.exe -k rps		Normal		
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 Process	SCardSvr	Stopped	Manual	Share	
	c:\windows\system32\scardsvr.exe			Ignore	
NT AUTHORITY\LocalService	0				
Task Scheduler Process	Schedule	Running	Auto	Share	
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Secondary Logon Process	seclogon	Running	Auto	Share	
	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) Process	SharedAccess	Stopped	Disabled	Share	
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share	
Share Process					
	c:\windows\system32\svchost.exe -k netsvcs		Ignore		
LocalSystem	0				
Print Spooler Process	Spooler	Stopped	Manual	Own	
	c:\windows\system32\spoolsv.exe		Normal		
LocalSystem	0				
Windows Image Acquisition (WIA) Disabled	stisvc	Stopped	Manual	Own	
Share Process					
	c:\windows\system32\svchost.exe -k imgs		Normal		
NT AUTHORITY\LocalService	0				
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual	Own	
Process					
	c:\windows\system32\svchost.exe -k swprv		Normal		
LocalSystem	0				
Performance Logs and Alerts	SysmonLog	Stopped	Manual	Own	
Process					
	c:\windows\system32\smlogsvc.exe		Normal		
NT Authority\NetworkService	0				
Telephony TapiSrv	Stopped	Manual	Share	Process	
	c:\windows\system32\svchost.exe -k tapisrv		Normal		

LocalSystem	0				
Terminal Services	TermService	Running	Manual		
Share Process					
	c:\windows\system32\svchost.exe -k termsvcs		Normal		
LocalSystem	0				
Themes	Themes	Stopped	Disabled	Share	Process
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Telnet	TlntSvr	Stopped	Disabled	Own	Process
	c:\windows\system32\tlntsvr.exe		Normal		NT
AUTHORITY\LocalService	0				
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled		
Share Process					
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Distributed Link Tracking Client	TrkWks	Running	Auto		
Share Process					
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Terminal Services Session Directory	Tssdis	Stopped	Disabled	Own	Process
	c:\windows\system32\tssdis.exe		Normal		
LocalSystem	0				
Windows User Mode Driver Framework	UMWdf	Stopped	Manual	Own	Process
	c:\windows\system32\wdmfrg.exe		Normal		
NT AUTHORITY\LocalService	0				
Upload Manager Process	uploadmgr	Stopped	Manual	Share	
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Uninterruptible Power Supply	UPS	Stopped	Manual	Own	Process
	c:\windows\system32\ups.exe		Normal		NT AUTHORITY\LocalService
LocalSystem	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	Stopped	Manual	Share	
Process					
	c:\windows\system32\svchost.exe -k localservice		Normal		NT AUTHORITY\LocalService
LocalSystem	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
LocalSystem	0				
WinHTTP Web Proxy Auto-Discovery Service					

WinHttpAutoProxySvc	Stopped	Manual			
Share Process					
	c:\windows\system32\svchost.exe -k localservice		Normal		NT AUTHORITY\LocalService
LocalSystem	0				
Windows Management Instrumentation	winmgmt	Running	Auto	Share	Process
	c:\windows\system32\svchost.exe -k netsvcs		Ignore		
LocalSystem	0				
Portable Media Serial Number Service	WmdmPmSN	Stopped	Manual	Share	Process
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Windows Management Instrumentation Driver Extensions	Wmi	Stopped	Manual	Share	Process
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own	Process
	c:\windows\system32\wbem\wmiapsrv.exe		Normal		
LocalSystem	0				
Automatic Updates Process	wuauerv	Stopped	Manual	Share	
	c:\windows\system32\svchost.exe -k netsvcs		Normal		
LocalSystem	0				
Wireless Configuration	WZCSVC	Stopped	Manual	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				
PsExec Process	PSEXESVC	Running	Manual	Own	
	c:\windows\psexecvc.exe		Ignore		
LocalSystem	0				
[Program Groups]					
Group Name	Name	User Name			
Accessories	Default	User:Accessories			Default
User					
Accessories\Accessibility		Default			
User:Accessories\Accessibility		Default User			
Accessories\Entertainment		Default			
User:Accessories\Entertainment		Default User			
Startup	Default User:Startup	Default User			
Accessories	All Users:Accessories	All Users			
Accessories\Accessibility		All			
Users:Accessories\Accessibility		All Users			
Accessories\Communications		All			
Users:Accessories\Communications		All Users			
Accessories\Entertainment		All			
Users:Accessories\Entertainment		All Users			
Accessories\System Tools		All Users:Accessories\System			

Tools All Users
 Administrative Tools All Users:Administrative Tools All Users
 Microsoft 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\Performance Tools All Users:Microsoft SQL Server 2005\Performance Tools All Users
 Startup All Users:Startup All Users
 Accessories NT AUTHORITY\SYSTEM:Accessories NT AUTHORITY\SYSTEM
 Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility NT AUTHORITY\SYSTEM
 Accessories\Entertainment NT AUTHORITY\SYSTEM:Accessories\Entertainment NT AUTHORITY\SYSTEM
 Startup NT AUTHORITY\SYSTEM:Startup NT AUTHORITY\SYSTEM
 Accessories FCLIENT38\Administrator:Accessories FCLIENT38\Administrator
 Accessories\Accessibility FCLIENT38\Administrator:Accessories\Accessibility FCLIENT38\Administrator
 Accessories\Entertainment FCLIENT38\Administrator:Accessories\Entertainment FCLIENT38\Administrator
 Administrative Tools FCLIENT38\Administrator:Administrative Tools FCLIENT38\Administrator
 Startup FCLIENT38\Administrator:Startup FCLIENT38\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	FCLIENT38\Administrator	Startup
Shortcut to BMListen	shortcut to bmlisten.lnk	FCLIENT38\Administrator	Startup
Shortcut to synctime	shortcut to synctime.lnk	FCLIENT38\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common Startup

[OLE Registration]

Object Local Server
 Sound (OLE2) sndrec32.exe
 Media Clip mplay32.exe
 Video Clipmplay32.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 Available Not
 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	6.0.3790.1830
Build	63790.1830
Application Path	C:\Program Files\Internet Explorer
Language	English (United States)
Active Printer	Not Available
Cipher Strength	128-bit
Content Advisor	Disabled
IEAK Install	No

[File Versions]

File	Version	Size	Date	Path
actxprxy.dll	6.0.3790.1830	97 KB	3/24/2005 5:55:26 PM	C:\WINDOWS\system32
advpack.dll	6.0.3790.1830	98 KB	3/24/2005 5:55:28 PM	C:\WINDOWS\system32
asctrls.ocx	6.0.3790.0	90 KB	3/25/2003 8:00:00 AM	C:\WINDOWS\system32

browsecl.dll	6.0.3790.0	62 KB	3/25/2003 8:00:00 AM	C:\WINDOWS\system32
browseui.dll	6.0.3790.1830	1,009 KB	3/24/2005 5:56:10 PM	C:\WINDOWS\system32
cdfview.dll	6.0.3790.1830	149 KB	3/24/2005 5:56:32 PM	C:\WINDOWS\system32
comctl32.dll	5.82.3790.1830	585 KB	3/24/2005 5:57:56 PM	C:\WINDOWS\system32
dxtrans.dll	6.3.3790.1830	205 KB	3/24/2005 6:00:58 PM	C:\WINDOWS\system32
dxtmsft.dll	6.3.3790.1830	355 KB	3/24/2005 6:00:58 PM	C:\WINDOWS\system32
iecont.dll	<File Missing>	Not Available	Not Available	Not Available
iecontlc.dll	<File Missing>	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.1830	324 KB	3/24/2005 6:04:58 PM	C:\WINDOWS\system32
iepeers.dll	6.0.3790.1830	248 KB	3/24/2005 6:04:58 PM	C:\WINDOWS\system32
iesetup.dll	6.0.3790.1830	61 KB	3/24/2005 6:04:58 PM	C:\WINDOWS\system32
ieuinit.inf	Not Available	24 KB	3/24/2005 6:04:58 PM	C:\WINDOWS\system32
ieexplore.exe	6.0.3790.1830	92 KB	3/24/2005 6:04:58 PM	C:\Program Files\Internet Explorer
imgutil.dll	6.0.3790.1830	38 KB	3/24/2005 6:05:04 PM	C:\WINDOWS\system32
inetcp1.cpl	6.0.3790.1830	358 KB	3/24/2005 6:05:06 PM	C:\WINDOWS\system32
inetcpl.dll	6.0.3790.0	109 KB	3/25/2003 8:00:00 AM	C:\WINDOWS\system32
inseng.dll	6.0.3790.1830	94 KB	3/24/2005 6:05:06 PM	C:\WINDOWS\system32

```

mlang.dll 6.0.3790.1830 578 KB 3/24/2005 6:07:20
PM C:\WINDOWS\system32 Microsoft
Corporation
msencode.dll 2002.10.4.0 112 KB
3/25/2003 8:00:00 AM
C:\WINDOWS\system32 呈口 物

mshta.exe 6.0.3790.1830 30 KB 3/24/2005 6:07:26
PM C:\WINDOWS\system32 Microsoft
Corporation
mshtml.dll 6.0.3790.1830 3,036 KB 3/24/2005 6:07:26
PM C:\WINDOWS\system32 Microsoft
Corporation
mshtml.tlb 6.0.3790.1830 1,320 KB 3/24/2005 6:07:26
PM C:\WINDOWS\system32 Microsoft
Corporation
mshtml.ed.dll 6.0.3790.1830 455 KB
3/24/2005 6:07:26 PM
C:\WINDOWS\system32 Microsoft
Corporation
mshtmlr.dll 6.0.3790.1830 56 KB
3/24/2005 6:07:26 PM
C:\WINDOWS\system32 Microsoft
Corporation
msident.dll 6.0.3790.1830 48 KB 3/24/2005 6:07:28
PM C:\WINDOWS\system32 Microsoft
Corporation
msidntld.dll 6.0.3790.0 15 KB 3/25/2003 8:00:00
AM C:\WINDOWS\system32 Microsoft
Corporation
msieftp.dll 6.0.3790.1830 244 KB 3/24/2005 6:07:28
PM C:\WINDOWS\system32 Microsoft
Corporation
msrating.dll 6.0.3790.1830 144 KB
3/24/2005 6:07:36 PM
C:\WINDOWS\system32 Microsoft
Corporation
mstime.dll 6.0.3790.1830 523 KB 3/24/2005 6:07:38
PM C:\WINDOWS\system32 Microsoft
Corporation
occache.dll 6.0.3790.1830 94 KB
3/24/2005 6:08:34 PM
C:\WINDOWS\system32 Microsoft
Corporation
proctexe.ocx 6.3.3790.1830 83 KB
3/24/2005 6:12:26 PM
C:\WINDOWS\system32 Intel Corporation

sendmail.dll 6.0.3790.1830 56 KB
3/24/2005 6:13:36 PM
C:\WINDOWS\system32 Microsoft
Corporation
shdoclc.dll 6.0.3790.0 589 KB 3/25/2003 8:00:00 AM
C:\WINDOWS\system32 Microsoft

```

```

Corporation
shdocvw.dll 6.0.3790.1830 1,468 KB
3/24/2005 6:13:36 PM
C:\WINDOWS\system32 Microsoft
Corporation
shfolder.dll 6.0.3790.1830 25 KB
3/24/2005 6:13:36 PM
C:\WINDOWS\system32 Microsoft
Corporation
shlwapi.dll 6.0.3790.1830 314 KB 3/24/2005 6:13:40
PM C:\WINDOWS\system32 Microsoft
Corporation
tdc.ocx 1.3.0.3130 58 KB 3/25/2003 8:00:00 AM
C:\WINDOWS\system32 Microsoft
Corporation
url.dll 6.0.3790.1830 37 KB 3/24/2005 6:26:12
PM C:\WINDOWS\system32 Microsoft
Corporation
urlmon.dll 6.0.3790.1830 673 KB 3/24/2005 6:26:12
PM C:\WINDOWS\system32 Microsoft
Corporation
webcheck.dll 6.0.3790.1830 273 KB
3/24/2005 6:26:16 PM
C:\WINDOWS\system32 Microsoft
Corporation
wininet.dll 6.0.3790.1830 646 KB 3/24/2005 6:26:18
PM C:\WINDOWS\system32 Microsoft
Corporation

[Connectivity]

Item Value
Connection Preference Never dial

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Disabled
AutoConfigURL
Proxy Disabled
ProxyServer
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

Transaction Manager Parameters

COM+ Settings

TPCC.AllTxns:

Activation:

Enable Object Pooling selected
Minimum Pool Size: <varies per client; see below>
Maximum Pool Size: <varies per client; see below>

Creating Timeout: 300,000
 Enable Object Construction
 Enable Just in Time Activation
 Concurrency:
 Concurrency Required

The Minimum and Maximum pool sizes were set to be equal to each other on each client.
 The values used varied per client as follows:
 - Clients 78 & 88 were set to 77
 - Client 18 was set to 68
 - All the remaining clients were set to 85

TPCC Application Registry Parameters

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\TPCC]
"Path"="C:\Inetpub\wwwroot\"
"NumberOfDeliveryThreads"=dword:0000000a
"MaxConnections"=dword:0000c350
"MaxPendingDeliveries"=dword:00001388
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="ibmserv4"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
"CallNoDuplicatesNewOrder"=dword:00000001
```

Microsoft Internet Information Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"PoolThreadLimit"=dword:00001ff8
"MaxPoolThreads"=dword:00000ffc
"ThreadTimeout"=dword:00015180
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"PerfIniFile"="infctrs.ini"
"Last Counter"=dword:000009a6
"Last Help"=dword:000009a7
"First Counter"=dword:00000966
"First Help"=dword:00000967
```

```
"Object List"="2406"
"Library Validation
Code"=hex:00,81,94,fc,3d,ba,c6,01,00,20,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:4c,c3,d3,e7,44,ca,56,e0,f3,e8,a0,14,52,26,fb,0f
"WbemAdapFileTime"=hex:d0,66,76,fb,3d,ba,c6,01
"WbemAdapFileSize"=dword:00002000
"WbemAdapStatus"=dword:00000000
"2003"=hex(b):d4,10,6f,ec,14,c3,c7,01
```

World Wide Web Service Registry Parameters

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,\
74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73,\
00,76,00,63,00,68,00,6f,00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00,\
6b,00,20,00,69,00,69,00,73,00,73,00,76,00,63,00,73,00,00,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):52,00,50,00,43,00,53,00,53,00,00,0,0,48,00,54,00,54,00,\
```

```
50,00,46,00,69,00,6c,00,74,00,65,00,72,00,00,00,49,00,49,00,53,00,41,00,44,\
00,4d,00,49,00,4e,00,00,00,00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet Information Services Manager"
"FailureActions"=hex:80,51,01,00,01,00,00,00,00,00,00,03,00,00,00,53,00,65,\
00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000006
"MinorVersion"=dword:00000000
"InstallPath"="C:\WINDOWS\system32\inetsrv"
"AccessDeniedMessage"="Error: Access is Denied."
"ServiceDll"=hex(2):43,00,3a,00,5c,00,57,00,49,00,4e,00,44,00,4f,00,57,00,53,\
00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,
```

```
69,00,6e,00,\
65,00,74,00,73,00,72,00,76,00,5c,00,69,00,69,00,73,00,77,00,33,00,61,00,64,\
00,6d,00,2e,00,64,00,6c,00,00,00
"AcceptExOutstanding"=dword:00000028
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Performance]
"Library"="C:\WINDOWS\system32\inetsrv\w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"PerfIniFile"="w3ctrs.ini"
"Last Counter"=dword:00000a9e
"Last Help"=dword:00000a9f
"First Counter"=dword:000009a8
"First Help"=dword:000009a9
"Object List"="2472 2646"
"Library Validation
Code"=hex:00,db,f6,fe,3d,ba,c6,01,00,5e,00,00,00,00,00,00
"WbemAdapFileSignature"=hex:39,e3,6c,2c,b4,be,59,f5,17,7c,c4,d5,2f,dc,f7,1a
"WbemAdapFileTime"=hex:90,a2,bd,fe,3d,ba,c6,01
"WbemAdapFileSize"=dword:00005e00
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,0,0,01,00,00,\
00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,\
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,00,\
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,
```

```
00,00,01,01,\
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,
00
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Service
s\W3SVC\Enum]
"0"="Root\LEGACY_W3SVC\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

RTE Input Parameters

IBM BenchMaster benchmark profile. DO NOT CHANGE THE SPACING IN THIS FILE!

108 ** Number of segments (all must be defined directly below)

SEGMENT	DB	MACHINE	LOG DIRECTORY	DB SERVER	SP
PREFIX	STARTWH	WEBSERVER	ENDWH	#USERS	
	GROUP1	GROUP2			
1A	frte15 1	c:\rtelogs 529	tpcc 5290	fclient15a n/a	ibmserv4 client15
1B	frte15 530	c:\rtelogs 1058	tpcc 5290	fclient15b n/a	ibmserv4 client15
1C	frte15 1059	c:\rtelogs 1587	tpcc 5290	fclient15c n/a	ibmserv4 client15
1D	frte15 1588	c:\rtelogs 2116	tpcc 5290	fclient15a n/a	ibmserv4 client15
1E	frte15 2117	c:\rtelogs 2645	tpcc 5290	fclient15b n/a	ibmserv4 client15
1F	frte15 2646	c:\rtelogs 3174	tpcc 5290	fclient15c n/a	ibmserv4 client15
1G	frte15 3175	c:\rtelogs 3703	tpcc 5290	fclient15a n/a	ibmserv4 client15

1H	frte15 3704	c:\rtelogs 4232	tpcc 5290	fclient15b n/a	ibmserv4 client15
1I	frte15 4233	c:\rtelogs 4765	tpcc 5330	fclient15c n/a	ibmserv4 client15
2A	frte18 4766	c:\rtelogs 5213	tpcc 4480	fclient18a n/a	ibmserv4 client18
2B	frte18 5214	c:\rtelogs 5661	tpcc 4480	fclient18b n/a	ibmserv4 client18
2C	frte18 5662	c:\rtelogs 6109	tpcc 4480	fclient18c n/a	ibmserv4 client18
2D	frte18 6110	c:\rtelogs 6557	tpcc 4480	fclient18a n/a	ibmserv4 client18
2E	frte18 6558	c:\rtelogs 7005	tpcc 4480	fclient18b n/a	ibmserv4 client18
2F	frte18 7006	c:\rtelogs 7453	tpcc 4480	fclient18c n/a	ibmserv4 client18
2G	frte18 7454	c:\rtelogs 7901	tpcc 4480	fclient18a n/a	ibmserv4 client18
2H	frte18 7902	c:\rtelogs 8349	tpcc 4480	fclient18b n/a	ibmserv4 client18
2I	frte18 8350	c:\rtelogs 8798	tpcc 4490	fclient18c n/a	ibmserv4 client18
3A	frte28 8799	c:\rtelogs 9313	tpcc 5150	fclient28a n/a	ibmserv4 client28
3B	frte28 9314	c:\rtelogs 9828	tpcc 5150	fclient28b n/a	ibmserv4 client28
3C	frte28 9829	c:\rtelogs 10343	tpcc 5150	fclient28c n/a	ibmserv4 client28
3D	frte28 10344	c:\rtelogs 10858	tpcc 5150	fclient28a n/a	ibmserv4 client28
3E	frte28 10859	c:\rtelogs 11373	tpcc 5150	fclient28b n/a	ibmserv4 client28
3F	frte28 11374	c:\rtelogs 11888	tpcc 5150	fclient28c n/a	ibmserv4 client28
3G	frte28 11889	c:\rtelogs 12403	tpcc 5150	fclient28a n/a	ibmserv4 client28

3H	frte28 12404	c:\rtelogs 12918	tpcc 5150	fclient28b n/a	ibmserv4 client28
3I	frte28 12919	c:\rtelogs 13436	tpcc 5180	fclient28c n/a	ibmserv4 client28
4A	frte35 13437	c:\rtelogs 13951	tpcc 5150	fclient35a n/a	ibmserv4 client35
4B	frte35 13952	c:\rtelogs 14466	tpcc 5150	fclient35b n/a	ibmserv4 client35
4C	frte35 14467	c:\rtelogs 14981	tpcc 5150	fclient35c n/a	ibmserv4 client35
4D	frte35 14982	c:\rtelogs 15496	tpcc 5150	fclient35a n/a	ibmserv4 client35
4E	frte35 15497	c:\rtelogs 16011	tpcc 5150	fclient35b n/a	ibmserv4 client35
4F	frte35 16012	c:\rtelogs 16526	tpcc 5150	fclient35c n/a	ibmserv4 client35
4G	frte35 16527	c:\rtelogs 17041	tpcc 5150	fclient35a n/a	ibmserv4 client35
4H	frte35 17042	c:\rtelogs 17556	tpcc 5150	fclient35b n/a	ibmserv4 client35
4I	frte35 17557	c:\rtelogs 18074	tpcc 5180	fclient35c n/a	ibmserv4 client35
5A	frte38 18075	c:\rtelogs 18584	tpcc 5100	fclient38a n/a	ibmserv4 client38
5B	frte38 18585	c:\rtelogs 19094	tpcc 5100	fclient38b n/a	ibmserv4 client38
5C	frte38 19095	c:\rtelogs 19604	tpcc 5100	fclient38c n/a	ibmserv4 client38
5D	frte38 19605	c:\rtelogs 20114	tpcc 5100	fclient38a n/a	ibmserv4 client38
5E	frte38 20115	c:\rtelogs 20624	tpcc 5100	fclient38b n/a	ibmserv4 client38
5F	frte38 20625	c:\rtelogs 21134	tpcc 5100	fclient38c n/a	ibmserv4 client38
5G	frte38 21135	c:\rtelogs 21644	tpcc 5100	fclient38a n/a	ibmserv4 client38

5H	frte38 21645 n/a	c:\rtelogs 22154	tpcc 5100	fclient38b n/a	ibmserv4 client38	7H	frte55 30919 n/a	c:\rtelogs 31442	tpcc 5240	fclient55b n/a	ibmserv4 client55	9H	frte68 40351 n/a	c:\rtelogs 40871	tpcc 5210	fclient68b n/a	ibmserv4 client68
5I	frte38 22155 n/a	c:\rtelogs 22662	tpcc 5080	fclient38c n/a	ibmserv4 client38	7I	frte55 31443 n/a	c:\rtelogs 31964	tpcc 5220	fclient55c n/a	ibmserv4 client55	9I	frte68 40872 n/a	c:\rtelogs 41392	tpcc 5210	fclient68c n/a	ibmserv4 client68
6A	frte48 22663 n/a	c:\rtelogs 23172	tpcc 5100	fclient48a n/a	ibmserv4 client48	8A	frte58 31965 n/a	c:\rtelogs 32491	tpcc 5270	fclient58a n/a	ibmserv4 client58	10A	frte75 41393 n/a	c:\rtelogs 41913	tpcc 5210	fclient75a n/a	ibmserv4 client75
6B	frte48 23173 n/a	c:\rtelogs 23682	tpcc 5100	fclient48b n/a	ibmserv4 client48	8B	frte58 32492 n/a	c:\rtelogs 33018	tpcc 5270	fclient58b n/a	ibmserv4 client58	10B	frte75 41914 n/a	c:\rtelogs 42434	tpcc 5210	fclient75b n/a	ibmserv4 client75
6C	frte48 23683 n/a	c:\rtelogs 24192	tpcc 5100	fclient48c n/a	ibmserv4 client48	8C	frte58 33019 n/a	c:\rtelogs 33545	tpcc 5270	fclient58c n/a	ibmserv4 client58	10C	frte75 42435 n/a	c:\rtelogs 42955	tpcc 5210	fclient75c n/a	ibmserv4 client75
6D	frte48 24193 n/a	c:\rtelogs 24702	tpcc 5100	fclient48a n/a	ibmserv4 client48	8D	frte58 33546 n/a	c:\rtelogs 34072	tpcc 5270	fclient58a n/a	ibmserv4 client58	10D	frte75 42956 n/a	c:\rtelogs 43476	tpcc 5210	fclient75a n/a	ibmserv4 client75
6E	frte48 24703 n/a	c:\rtelogs 25212	tpcc 5100	fclient48b n/a	ibmserv4 client48	8E	frte58 34073 n/a	c:\rtelogs 34599	tpcc 5270	fclient58b n/a	ibmserv4 client58	10E	frte75 43477 n/a	c:\rtelogs 43997	tpcc 5210	fclient75b n/a	ibmserv4 client75
6F	frte48 25213 n/a	c:\rtelogs 25722	tpcc 5100	fclient48c n/a	ibmserv4 client48	8F	frte58 34600 n/a	c:\rtelogs 35126	tpcc 5270	fclient58c n/a	ibmserv4 client58	10F	frte75 43998 n/a	c:\rtelogs 44518	tpcc 5210	fclient75c n/a	ibmserv4 client75
6G	frte48 25723 n/a	c:\rtelogs 26232	tpcc 5100	fclient48a n/a	ibmserv4 client48	8G	frte58 35127 n/a	c:\rtelogs 35653	tpcc 5270	fclient58a n/a	ibmserv4 client58	10G	frte75 44519 n/a	c:\rtelogs 45039	tpcc 5210	fclient75a n/a	ibmserv4 client75
6H	frte48 26233 n/a	c:\rtelogs 26742	tpcc 5100	fclient48b n/a	ibmserv4 client48	8H	frte58 35654 n/a	c:\rtelogs 36180	tpcc 5270	fclient58b n/a	ibmserv4 client58	10H	frte75 45040 n/a	c:\rtelogs 45560	tpcc 5210	fclient75b n/a	ibmserv4 client75
6I	frte48 26743 n/a	c:\rtelogs 27250	tpcc 5080	fclient48c n/a	ibmserv4 client48	8I	frte58 36181 n/a	c:\rtelogs 36703	tpcc 5230	fclient58c n/a	ibmserv4 client58	10I	frte75 45561 n/a	c:\rtelogs 46081	tpcc 5210	fclient75c n/a	ibmserv4 client75
7A	frte55 27251 n/a	c:\rtelogs 27774	tpcc 5240	fclient55a n/a	ibmserv4 client55	9A	frte68 36704 n/a	c:\rtelogs 37224	tpcc 5210	fclient68a n/a	ibmserv4 client68	11A	frte78 46082 n/a	c:\rtelogs 46540	tpcc 4590	fclient78a n/a	ibmserv4 client78
7B	frte55 27775 n/a	c:\rtelogs 28298	tpcc 5240	fclient55b n/a	ibmserv4 client55	9B	frte68 37225 n/a	c:\rtelogs 37745	tpcc 5210	fclient68b n/a	ibmserv4 client68	11B	frte78 46541 n/a	c:\rtelogs 46999	tpcc 4590	fclient78b n/a	ibmserv4 client78
7C	frte55 28299 n/a	c:\rtelogs 28822	tpcc 5240	fclient55c n/a	ibmserv4 client55	9C	frte68 37746 n/a	c:\rtelogs 38266	tpcc 5210	fclient68c n/a	ibmserv4 client68	11C	frte78 47000 n/a	c:\rtelogs 47458	tpcc 4590	fclient78c n/a	ibmserv4 client78
7D	frte55 28823 n/a	c:\rtelogs 29346	tpcc 5240	fclient55a n/a	ibmserv4 client55	9D	frte68 38267 n/a	c:\rtelogs 38787	tpcc 5210	fclient68a n/a	ibmserv4 client68	11D	frte78 47459 n/a	c:\rtelogs 47917	tpcc 4590	fclient78a n/a	ibmserv4 client78
7E	frte55 29347 n/a	c:\rtelogs 29870	tpcc 5240	fclient55b n/a	ibmserv4 client55	9E	frte68 38788 n/a	c:\rtelogs 39308	tpcc 5210	fclient68b n/a	ibmserv4 client68	11E	frte78 47918 n/a	c:\rtelogs 48376	tpcc 4590	fclient78b n/a	ibmserv4 client78
7F	frte55 29871 n/a	c:\rtelogs 30394	tpcc 5240	fclient55c n/a	ibmserv4 client55	9F	frte68 39309 n/a	c:\rtelogs 39829	tpcc 5210	fclient68c n/a	ibmserv4 client68	11F	frte78 48377 n/a	c:\rtelogs 48835	tpcc 4590	fclient78c n/a	ibmserv4 client78
7G	frte55 30395 n/a	c:\rtelogs 30918	tpcc 5240	fclient55a n/a	ibmserv4 client55	9G	frte68 39830 n/a	c:\rtelogs 40350	tpcc 5210	fclient68a n/a	ibmserv4 client68	11G	frte78 48836 n/a	c:\rtelogs 49294	tpcc 4590	fclient78a n/a	ibmserv4 client78

Appendix D: 60-Day Space

Note : Numbers are in KBytes unless otherwise specified

Warehouses	54349	tpmC	684508	tpmC/W	12.59	
Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	56,000	5,976	136	305.60		6,417.60
District	560,000	62,224	232	3,122.80		65,578.80
Item	100,000	9,416	136	477.60		10,029.60
New-order	504,000,000	8,979,960	22,632		4,480,000	13,482,592.00
History	1,680,000,000	98,102,192	366,512		19,257,890.18	117,726,594.18
Orders	1,680,000,000	54,857,144	26,758,824		15,961,938.01	97,577,906.01
Customer	1,680,000,000	1,221,818,184	76,233,280	64,902,573.20		1,362,954,037.20
Order-line	16,799,949,701	1,101,636,048	2,594,680		215,958,504.90	1,320,189,232.90
Stock	5,600,000,000	1,792,000,000	3,778,592	89,788,929.60		1,885,567,521.60
Totals		4,277,471,144	109,755,024	154,695,408.80	255,658,333.09	4,797,579,909.89
Segment	LogDev Cnt.	Seg. Size	Needed	Overhead	Not Needed	
misc	14	1,792,000,000	1,549,058,351	15,490,584	227,451,065.40	
big	14	3,297,280,000	3,248,521,559	32,485,216	16,273,225.61	
master.msdb.model	1	13,312	13,312		-	
tpcc_root	1	8,192	8,192		-	
tempdb	1	8,704	8,704		-	
Totals		5,089,310,208.00	4,797,610,117.89	47,975,799.10	243,724,291.01	
Dynamic space	1,254,595,384.00	Sum of Data for Order, Order-Line and History				
Static space	3,335,301,991.90	Data + Index + 5% Space + Overhead - Dynamic space				
Free space	255,688,541.09	Total Seg. Size - Dynamic Space - Static Space - Not Needed				
Daily growth	245,365,879.17	Dynamic space/(Wc * 62.5) * tpmC				
Daily spread	(112,360,277.67)	Free space - 1.5 * Daily growth (zero if negative)				
60 day (KB)	18,057,254,742.37	Static space + 60 * (daily growth + daily spread)				
60 day (GB)	17,220.74	60-day space in GB (excludes OS, Paging and RDBMS logs)				
Available (GB)	91,741.44	The total storage configured and available for the database, minus logs, in a RAID-0 configuration				

Log File Storage Requirement:

Log size (MB)	2,300,000.00	Total size of log file
% Log Used	32.1419	% of log file used during entire run
Total N-O Txn	118,413,231	Total count of new order transactions during entire run
Log per N-O txn	6.39	KB of log storage used per New Order transaction
8 Hour Log (GB)	2,003.18	8-hour log storage required
Log configured (GB)	3,726.09	

Disk Capacity	MB	GB
36GB-15K RPM	34,204	33.40
73.4GB-15K RPM	69,898	68.26
500GB-7.2K RPM	476,940	465.76

Space Usage	GB Needed	Disks Priced	GB Priced	GB Usable	TB Usable
60-Day (RAID-0)	17,220.74	1,344 73GB-15K RPM	91,741.44	91,741.44	89.5913
			Total DB	91,741.44	89.5913
8hr Log (RAID-10)	2,003.18	16 500GB-7.2K RPM	7,452.19	3,726.09	3.6388
			Total Log	3,726.09	3.6388
OS, SQL Server	8.00	1 73GB-15K RPM	68.26	68.26	0.0667
Total Space	19,231.92	1361	99,261.89	95,535.79	93.29

Appendix E: Third-Party Price Quotes

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
<http://www.microsoft.com/>

Microsoft

August 26, 2008

IBM Corporation
Chris King
3079 Cornwallis Road
Durham, NC 27709

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	4	\$93,728
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	12	\$8,628
P72-01684	Windows Server 2003 R2 Enterprise x64 Edition <i>Server License Only - No CALs</i> <i>Discount Schedule: Open Program - No Level</i> <i>Unit Price reflects a 42% discount from the retail unit price of \$3,999.</i>	\$2,334	1	\$2,334
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

Windows Server 2008 and Windows Server 2003 are currently orderable through Microsoft's normal distribution channels. A list of Microsoft's resellers can be found at <http://www.microsoft.com/products/info/render.aspx?view=22&type=mp&content=22/licensing>

SQL Server 2008 will be orderable and available by August 30, 2008.

Defect support is included in the purchase price. Additional support is available from Microsoft PSS on an incident by incident basis at \$245 per call.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCchki0808260000009757.



Please include this Reference ID in any correspondence regarding this price quote.



[Home](#) > [My Shopping Cart](#)

MY SHOPPING CART

[My Wish Lists](#) | [Print Cart](#) | [Email Cart](#)

<input type="checkbox"/>	Qty.	Product Description	Savings	Total Price
<input type="checkbox"/>	1	 NETGEAR JGS524 10/100/1000Mbps Gigabit Ethernet Switch - Retail Item #: N82E16833122058 Return Policy: Limited 30-Day Return Policy	\$40.00 Mail-in Rebate	\$224.99
<input type="button" value="Protect Your Investment (expand for options)"/>				
<input type="checkbox"/>	1	 BELKIN A3L791-10-BLK 10 ft. Cat 5E Black RJ45 CAT5e Patch Cable - Retail Item #: N82E16812106332 Return Policy: Standard Return Policy		\$4.49
Subtotal:				\$229.48

Calculate Shipping

Zip Code:

Shipping: \$0.00

Redeem Gift Certificates

Claim Code:
 Security Code:

Gift Certificates: \$0.00

Apply Promo Code

Promo Code:

Promo Code: \$0.00

Grand Total:* \$229.48

* Above total does not include shipping or taxes. Please input zip code to calculate your grand total.

Having problems with your cart? [Click here](#) for help or try [emptying your cart to start over](#).

▶ [Click here to view important shipping information.](#)

[Policy & Agreement](#) | [Privacy Policy](#) | © 2000-2008 Newegg Inc. All rights reserved.