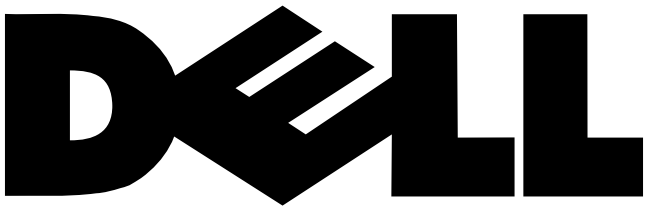


TPC Benchmark™ C
Full Disclosure Report
for
Dell PowerEdge 2900
using
Microsoft SQL Server 2005 Standard x64 Edition and
Microsoft Windows Server 2003 Standard x64 Edition SP1

Second Edition
Submitted for Review
March 9, 2007.
Updated April 23, 2007 for pricing.

Second Edition, April 23, 2007

Dell believes that the information included in this document is accurate as of the publication date. The information in this document is subject to change without notice. Furthermore, Dell is not responsible for any errors contained within this document.

The pricing information given in this FDR is accurate as of the second publication date, April 23, 2007 and is generally available.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Actual performance experienced by a particular customer may vary due to differences in system layout and configuration, hardware and/or software revision levels, and background system activity. The content of this document is for informational purposes only.

PowerEdge and PowerVault are registered trademarks of Dell Inc.

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

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

Intel®, and Xeon® are registered trademarks of Intel Corporation.

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark™ C test conducted on Dell PowerEdge 2900. The tests were run in a client/server configuration using one PowerEdge SC 1430 as client. The operating system used for the benchmark was Microsoft Windows Server 2003 SP1, Microsoft SQL Server 2005 Standard x64 Edition on the database server and Microsoft Windows Server 2003 Standard Edition on the client. The database was Microsoft SQL Server 2005 Standard x64 Edition. Microsoft COM+ provided the database connection queues. All tests were done in compliance with Revision 5.8 of the Transaction Processing Council's TPC Benchmark™ C Standard Specification. Two standard TPC Benchmark™ C metrics, transactions per second (tpmC) and price per tpmC ($/tpmC) are reported and referred to in this document. The results from the tests are summarized below.

Hardware	Software	Total System Cost	tpmC	$/tpmC	Availability Date
Dell PowerEdge 2900	Microsoft SQL Server 2005 Standard x64 Edition With Windows Server 2003 Standard x64 Edition SP1	$63,080	69,564	$.91	March 9, 2007

Auditor

The results of the benchmark and test methodology used to produce the results were audited by Lorna Livingtree of Performance Metrics and have fully met the TPC-C rev 5.8 specifications.

Additional copies of this Full Disclosure Report can be obtained from either the Transaction Processing Performance Council or Dell at the following address:

Transaction Processing Performance Council (TPC)
c/o Adminstrator, TPC
Presidio of San Francisco
Bldg 572B Ruger St.
San Francisco, CA 94129-0920
Phone: (415) 561-6272, fax 415-561 6120
www.tpc.org

or

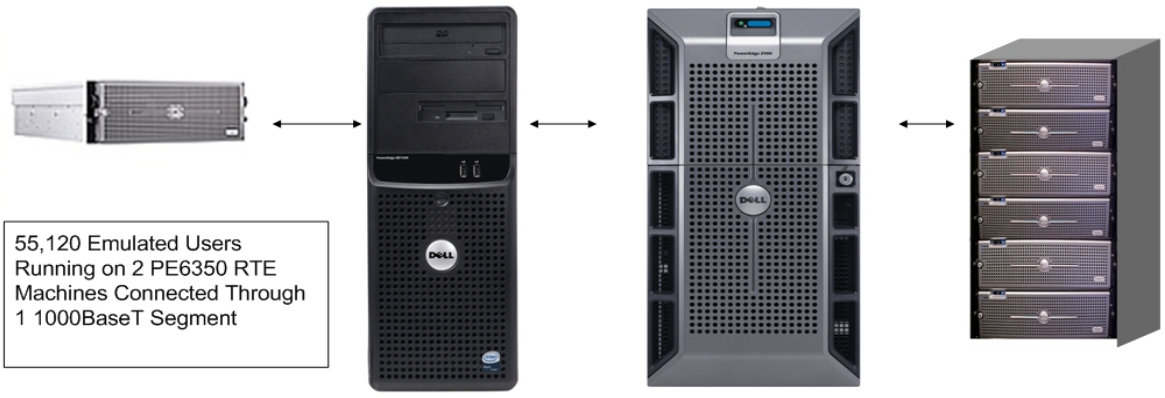
Dell
1 Dell Way
Round Rock, TX 78682
Attention: Mike Molloy, Ph.D.

DELL™	**PowerEdge 2900 Server with PowerEdge SC1430 Client**		TPC-C Rev 5.8 Original Report Date **March 9, 2007** **Updated April 23, 2007 for pricing**
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$63,080	**69,564 tpmC**	**$.91 / tpmC**	March 9, 2007

Processor	Database Manager	Operating System	Other Software	Number of Users
1/4/4 Quad Core Intel® Xeon® E5345, 2X4MB Cache, 2.33GHZ 1333, 667MHZ FSB	Microsoft SQL Server 2005 Standard x64 Edition	Microsoft Windows Server 2003 Standard x64 Edition SP1	Windows Server 2003 Standard Edition w/ COM+ Internet Information Server 6.0 Microsoft Visual C++	55,120

PowerEdge SC1430 Client
2/2/4 5140, Dual Core Intel® Xeon®, 4MB cache, 2.33GHz, 1333MHz FSB
1024 MB RAM
Broadcom NetXtreme II Gigabit Adapter
1 80GB SATA 7.2K Disk

PowerEdge 2900
1/4/4 E5345, Quad Core Intel® Xeon® , 2x4MB cache, 2.33GHz, 1333MHz FSB
24GB 667MHz (12x2GB), Dual Ranked Fully Buffered DIMM
Dual embedded Broadcom NetXtreme II Gigabit Ethernet NICs
1 PERC5i SAS RAID Controller
3 Dell PERC5e SAS RAID Controllers
8 146GB,3GBPS,SAS,3.5IN 10K



55,120 Emulated Users
Running on 2 PE6350 RTE Machines Connected Through 1 1000BaseT Segment

6 PowerVault MD1000 SAS Enclosures
90 36GB, 3GBPS, SAS, 3.5IN 15K

System Component	Server		Each Client	
Processor/Core/Cache	1	1/4/4 Quad Core Intel® Xeon® E5345, 2.33GHz, 1333, 2X4MB	2	2/2/4 Dual Core Intel® Xeon® 5140 2.33Ghz 4MB L2
Memory		24GB 667 FB-DIMM		1024 MB
Disk Controllers	3 1	Dell PERC5 RAID Controllers Integrated PERC5i Raid Controller.	1	Onboard SATA
Disk Drives	90 8	36GB SAS 15K 146GB SAS 10K	1	80GB 7.2K SATA
Total Storage	98	3548GB SAS	1	80GB SATA
Other	2 1	Broadcom NetXtreme II GigE CD-ROM	2 1	Broadcom NetXtreme II GigE CD-ROM

Dell	PowerEdge 2900				TPC-C REV 5.8 EXECUTIVE SUMMARY		
				Report Date: March-09 Updated April 23, 2007 for Pricing.			
Description	Part Number	Third Party		Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware		Brand	Pricing				
5345,2X4MB/2.33GHz,1333FSB	222-7261			$1,542.00	1	$1,542.00	$320.00
24GB 667MHz(12x2GB),2R	311-6201			$4,618.00	1	$4,618.00	
PERC5/E,SAS,EXT,PCI-E,MD1000	341-3023			$799.00	3	$2,397.00	
146GB SAS 10K (OS+LOG)	341-3029			$369.00	8	$2,952.00	
E773s 17-inch Color CRT Monitor	E7733YR			$149.00	1	$149.00	
					Subtotal	$11,658.00	$320.00
PowerVault Disk Subsystem							
PV MD1000,RACK,3U,15 BAY,LBZL	220-4476			$2,680.00	6	$16,080.00	$9,888.00
SINGLE ENCL MGT MODULE, SAS ONLY	420-5927			$345.83	6	$2,074.98	
SAS CABLE,1M,MD1000	310-7082			$30.00	6	$180.00	
36GB SAS 15K (Data+Backups)	340-9472			$249.00	90	$22,410.00	
Dell Depth 4 Post Rack 30U	RACK-111-30-D	Racksolutions.com	3	$429	1	$429.00	
					Subtotal	$41,173.98	$9,888.00
Server Software							
SQL Server 2005 Std x64 Edition, Per processor licensing *	228-03128	Microsoft.com	1	$5,999.00	1	$5,999.00	
Windows Server 2003 Standard x64 Edition **	P73-0295	Microsoft.com	1	$719.00	1	$719.00	
Professional Support (1 Incident)		Microsoft.com	1	$245.00	1		$245.00
					Subtotal	$6,718.00	$245.00
Client Hardware							
Dell PowerEdge SC 1430, 2.33GHZ/4MB,1333 FSB	222-3183			$1,060.00	1	$1,060.00	$320.00
Additional processor, 5140,4MB/2.33GHz,1333FSB	311-6142			$749.00	1	$749.00	
1GB,667MHz,(2X512MB),1R,FBD	311-6151			$339.00	1	$339.00	
80GB,SATA,1IN,7.2K RPM,HD ,7.2K	341-3757			$99.00	1	$99.00	
BCOM NetX 5721,Gb,ETHERNET,NIC	430-1496			$59.00	1	$59.00	
E773s 17-inch Color CRT Monitor	E7733YR			$149.00	1	$149.00	
					Subtotal	$2,455.00	$320.00
Client Software							
Windows Server 2003 Standard Edition **	P73-00295	Microsoft.com	1	$719.00	1	$719.00	
Visual C++ Standard Edition	254-00170	Microsoft.com	1	$109.00	1	$109.00	
					Subtotal	$828.00	
User Connectivity							
5ft Crossover cable	CBLC5C7	LanAdapter.com	2	$1.38	3	$4.14	
					Subtotal	$4.14	
All Hardware and maintence components from Dell are discounted 16% based on total dollar value of this configuration.					Other Discounts	$10,530.40	
					Total USD:	$52,307	$10,773
Notes: For pricing verification call 1-800-BUY-Dell and reference Quote# 302841264 as a complex quote.				Three-Year Cost of Ownership USD:			$63,080
** All Microsoft maintenance is covered by the maintenance costs of Microsoft SQL Server							
Pricing: 1 - Microsoft 2 - LanAdapter.com 3 -RackSolutions.com						tpmC Rating:	69564
Audited by Lorna Livingtree, Performance Metrics Inc.						USD$ / tpmC:	0.91

Prices used in the 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. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark pricing specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org.

MQTh, computed Maximum Qualified Throughput 69,564 tpmC

Response Times (in seconds)	Average	90th	Max
- Neworder	0.19	0.26	5.00
- Payment	0.13	0.15	1.66
- Delivery (interactive portion)	0.10	0.11	1.26
- Stock-Level	0.22	0.30	1.58
- Order Status	0.18	0.25	4.68
- Delivery (deferred portion)	0.17	0.25	5.03
- Menu	0.11	0.11	1.75

Response time delay added for emulated components Menu 0.1
Resp 0.1

Transaction Mix, in percent of total transactions

- New-Order	44.82%
- Payment	43.03%
- Delivery	4.05%
- Stock-Level	4.05%
- Order-Status	4.04%

Keying/Think Times (in seconds),	Min		Average		Max	
- New-Order	18.03	0.0	18.03	12.05	18.94	120.44
- Payment	3.03	0.0	3.03	12.06	3.95	120.43
- Delivery	2.03	0.0	2.03	5.07	2.94	50.42
- Stock-Level	2.03	0.0	2.03	5.07	2.94	50.43
- Order-Status	2.03	0.0	2.03	10.05	2.94	100.43

Test Duration

- Ramp-up time	68 minutes
- Measurement interval	120 minutes
- Number of checkpoints	4
- Checkpoint interval	30 minutes
- Number of transactions (all types)	19,382,105

Table of Contents

Document Structure

The TPC Benchmark C Standard Specification Revision 5.8, written and approved by the Transaction Processing Performance Council (TPC), determines the contents of this report. The format of this report is based on this specification. Most sections of this report begin with the specification requirements printed in italic type, immediately followed by the detail in plain type of how Dell complied with the specification. Where extensive listings are required (such as listing of code), a note is included which references an appendix containing the listing.

Benchmark Overview

TPC Benchmark™ C (TPC-C) is an OLTP workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

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

The performance metric reported by TPC-C is a "business throughput" measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint.

The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

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

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

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of

systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

System Overview

The hardware configuration used in this TPC-C test is a Dell PowerEdge 2900 server driven by one Dell PowerEdge SC1430 client. The client and server are networked together via ethernet cables. Two remote terminal emulators (RTE) systems PowerEdge 6350 emulate users executing the standard TPC-C workload. The RTE are connected to the client through 1000 BaseT segments. The segment connects to the client machine at 1000 BaseT and to the RTE machine at 1000Mbit/sec, full duplex. Microsoft Windows Server 2003, Standard x64 Edition SP1 was the operating system used on the server. Microsoft Windows Server 2003, Standard x64 Edition was used on the client. Microsoft SQL Server 2005 Standard x64 Edition was the database on the server machine.

The PowerEdge 2900 motherboard uses an Intel chipset and can hold up to 2/4/4 Xeon® Quad Core processors (2.66GHz Max). This result used 1/4/4/ 2.33 GHz 1333 with 2 x 4MB L2 cache and 64-bit Extensions. The system has 1 PCI-e x8 slot, 3 PCI-e x4 slots, and 2 PCI-x 64 Bit/133 MHz slots. The measured configuration used 24 GB of fully buffered 667 DDR2 RAM, which was achieved using 12 2 GB DIMMs. The network adapters are embedded with Broadcom NetXtreme II GigE network adapters.

The PowerEdge 2900 has an integrated 6 slot riser board to which was attached 8 146GB SAS LFF disks in RAID 10 configuration containing the database log and operating system via an internal channel on a Dell internal PERC5i RAID controller. In addition, three Dell PERC5e PCI-e RAID controllers were installed in PCI-e slots for the data volumes. The Dell PERC5e PCI-e RAID controllers were connected to 6 MD1000 disk pods enclosing a total of 90 36GB 15K RPM SAS disks.

The client has 2/2/4 2.33GHz Intel Xeon® processors with 4MB of L2 cache. The client has 1024 Mbytes of RAM, one 80 GB hard disk, 2 Broadcom NetXtreme II network adapters, one embedded.The client's Broadcom adapters were connected to the RTE machine and database server through ethernet cables.

General Items

Test Sponsor

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

Dell was the test sponsor of this TPC Benchmark™ C.

Application Code and Definition Statements

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

The application consists of the Microsoft Benchcraft Remote Terminal Emulator (RTE) program emulating a set of users entering TPC-C transactions through web browsers, and communicating with Client machines running the Microsoft Internet Information Server (IIS) web server. The Client machines use the COM+ transaction monitor to communicate with the database server machine.

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

The web Client code is listed in Appendix A.

Parameter Settings

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

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

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

Appendix C contains all the database, Windows 2003 Server, and Internet Information Service parameters used in this benchmark.

Appendix D contains the 60 day space calculations.

Configuration Diagrams

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

Figures 1 and 2 respectively show the measured and priced full client/server configurations. The system under test (SUT) in the measured system was identical to what was priced.

Figure 1: Measured Configuration

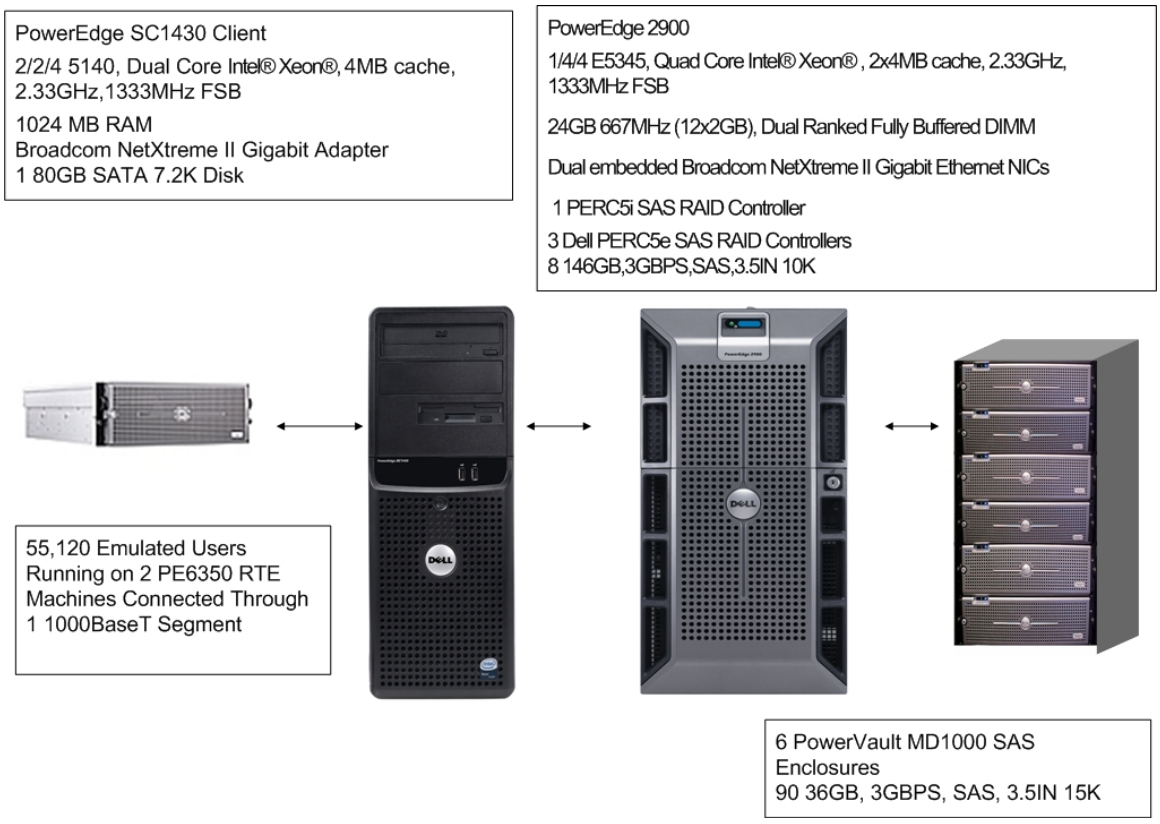
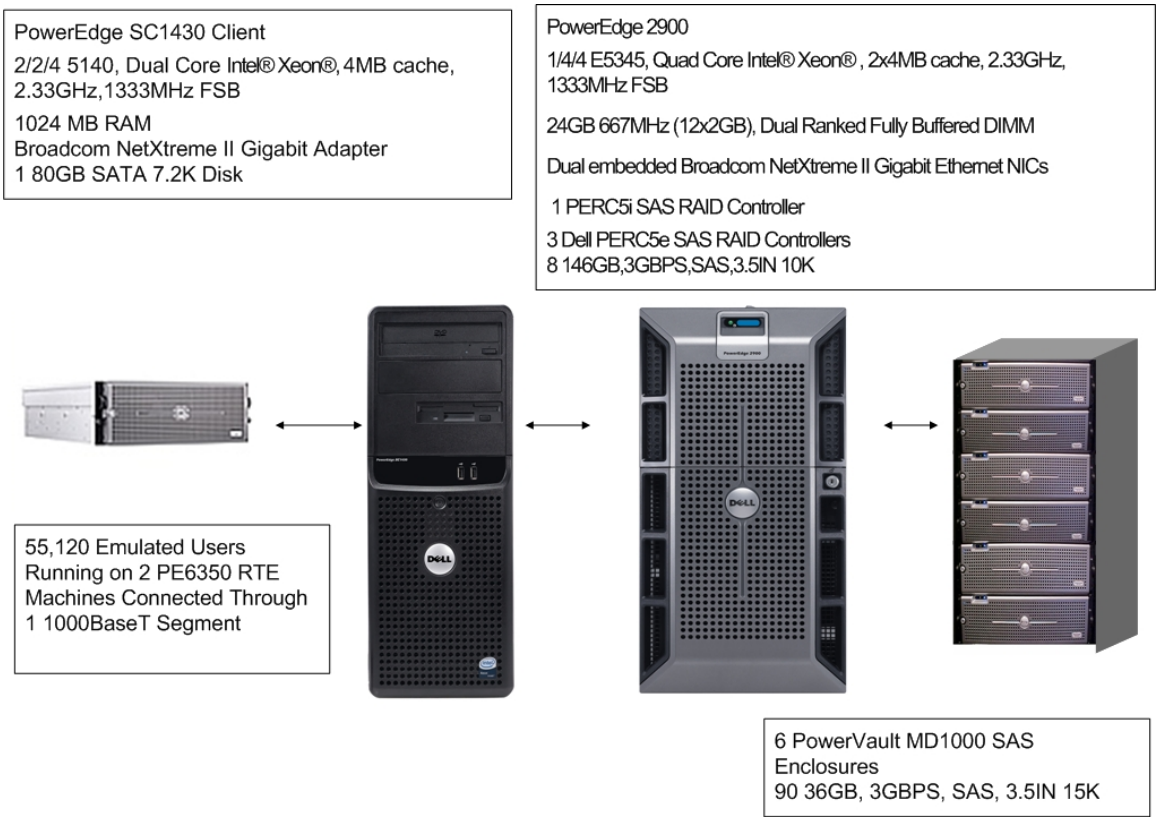


Figure 2: Priced Configuration



Clause 1 -- Logical Database Design Related Items

Table Definitions

Listings must be provided for all table definition statements and all other statements used to set-up the database. (8.1.2.1)

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

Physical Organization of the Database

The physical organization of tables and indices, within the database, must be disclosed. (8.1.2.2)

The measured configuration used 98 disk drives. The organization is shown in Table 5: Data Distribution.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any 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. (8.1.2.3)

Insert and delete functionality was fully operational during the benchmark.

Horizontal and Vertical Partitioning

While there are a 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. (8.1.2.4)

Partitioning was not used in this benchmark.

Replication

Replication of tables, if used, must be disclosed (see Clause 1.4.6). (8.1.2.5)

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). (8.1.2.6)

No additional attributes were used in this benchmark.

Clause 2 -- Transaction and Terminal Profiles Related Items

Random Number Generation

The method of verification for the random number generation must be described. (8.1.3.1)

The random number generation was done internal to the Microsoft BenchCraft RTE program, which was audited independently. The benchcraft RTE from Microsoft computes random integers as described in "Random Number Generators: Good Ones are Hard to Find." Communications of the ACM – October 1988 Volume 31 Number 10.

Screen Layout

The actual layouts of the terminal input/output screens must be disclosed. (8.1.3.2)

The screen layouts are based on those in Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C Standard Specification. There are some very minor differences based on the fact that this is a web client implementation.

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

The terminal features were verified by allowing the auditor to manually execute each of the five transaction types, using Microsoft Internet Explorer version 3.0.

Intelligent Terminals

Any usage of presentation managers or intelligent terminals must be explained. (8.1.3.4)
Comment 1*: The intent of this clause is to describe any special manipulations performed by a local terminal or workstation to off-load work from the SUT. This includes, but is not limited to: screen presentations, message bundling, and local storage of TPC-C rows.*
Comment 2*: This disclosure also requires that all data manipulation functions performed by the local terminal to provide navigational aids for transaction(s) must also be described. Within this disclosure, the purpose of such additional function(s) must be explained.*

Application code involved in the manipulation of data was run on the client. Screen manipulation commands in the form of HTML were downloaded to the web browser, which handled input and output presentation graphics. A listing of this code is included in Appendix A. Microsoft Internet Information Service assisted in the processing and presentation of this data.

Transaction Profiles

The percentage of home and remote order-lines in the New-Order transactions must be disclosed. (8.1.3.5)
The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed. (8.1.3.6)
The number of items per orders entered by New-Order transactions must be disclosed. (8.1.3.7)
The percentage of home and remote Payment transactions must be disclosed. (8.1.3.8)
The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the database must be disclosed. (8.1.3.9)
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. (8.1.3.10)

Table 1: Transaction Statistics

Transaction	Function	Value
New Order	Home Warehouse Items	99.00%
	Remote Warehouse Items	1.00%
	Rolled Back Transactions	1.00%
	Average Lines Per Order	10.00
Payment	Home Warehouse	85.00%
	Remote Warehouse	0.15%
	Non-Primary Key Access	60.03%
Order Status	Non-Primary Key Access	60.01%
Delivery	Skipped Transactions	0

Transaction Mix

The mix (i.e., percentages) of transaction types seen by the SUT must be disclosed. (8.1.3.11)

Table 2: Transaction mix

Transaction	Percentage
New Order	44.82%
Payment	43.04%
Order Status	4.05%
Delivery	4.05%
Stock Level	4.04%

Deferred Delivery Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed. (8.1.3.12)

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

The source code is listed in Appendix A.

Clause 3 -- Transaction and System Properties Related Items

ACID Tests

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7. (8.1.4.1)

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

Atomicity

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

Completed Transactions

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

Aborted Transactions

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

Consistency

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

Consistency conditions one through four were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests. A run was executed under full load lasting over ten (10) minutes and included a checkpoint. The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

Isolation

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

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

Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer, and the order returned was noted. T0 was committed

2. A New-Order transaction T1 was started for the same customer used in T0. T1 was stopped prior to the commit.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to complete and was committed.
5. An Order-status transaction T3 was started for the same customer used in T1. T3 returned the order inserted by T1.

   This outcome demonstrates serialization of T2 before T1. It has the equivalent validity to the outcome specified in the Standard which supposes T1 to be serialized before T2.

Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-status and New-Order transactions when the New-Order transaction is rolled back.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T0 was executed for a randomly selected customer and the order returned was noted. T0 was committed.
2. A New-Order transaction T1 with an invalid item number, was started for the same customer used in T0. T1 was stopped immediately prior to rollback.
3. An Order-Status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.
4. T1 was allowed to rollback.
5. An Order-status transaction T3 was started for the same customer used in T1. T3 returned the same order that T0 had returned.

Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions.

The execution of the above test proceeded as follows:

1. The D_Next_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1 was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to commit.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to complete. T2 completed and was committed.
5. The order number returned by T1 was the same as the D_Next_O_ID retrieved in step 1. The order number returned by T2 was one greater that the order number returned by T1.
6. The D_Next_O_ID of the same district was retrieved again. It has been incremented by two (i.e. it was one greater than the order number returned by T2).

Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is rolled back.

The execution of the above test proceeded as follows:

1. The D_Next_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1, with an invalid item number, was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to rollback.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to roll back, and T2 completed and was committed.
5. The order number returned by T2 was the same as the D_Next_O_ID retrieved in step 1.
6. The D-Next_O_ID of the same district was retrieved again. It has been incremented by one (i.e. one greater that the order number returned by T2).

Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_Balance of the customer found in step 1 was retrieved.
3. A delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the commit of the database transaction cooresponding to the district used in step 1.
4. A payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to complete. T2 completed and committed.
6. The C_Balance of the customer found in step 1 was retrieved again. The C_Balance reflected the results in both T1 and T2.

Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is rolled back.

The execution of the above test proceeded as follows:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_Balance of the customer found in step 1 was retrieved.
3. A delivery business transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the roll back of the database transaction cooresponding to the district used in step 1.
4. A payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to rollback. T2 completed and committed.
6. The C_Balance of the customer found in step 1 was retrieved again. The C_Balance reflected the results of only T2.

Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the price of an item.

The execution of the above test proceeded as follows:

1. The I_Price of two randomly selected items X and Y were retrieved.
2. A New-Order transaction T2 with a group of items X and Y was started. T2 was stopped immediately after retrieving the prices of all items. The prices of itmes X and Y retrieved matched those in step 1.
3. A transaction T3 was started to increase the price of items X and Y by 10%.
4. T3 did not stall and no transaction was rolled back. T3 was committed.
5. T2 was resumed, and the prices of all items were retrieved again within T2. The prices of items X and Y matched those retrieved in step 1.
6. T2 was committed.
7. The prices of items X and Y were retrieved again. The values matched the values set by T3.

Execution followed Case D of Clause 3.4.2.7.

Isolation Test 8

This test demonstrates isolation for phantom protection between New-Order and Order-Status transactions.

The execution of the above test proceeded as follows:

1. An Order-Status transaction T1 was started for a randomly selected customer.
2. T1 was stopped immediately after reading the order table for the selected customer. The most recent order for that customer was found.

3. A New-Order transaction T2 was started for the same customer. T2 completed and was committed without being blocked by T1.
4. T1 was resumed and the order table was read again to determine the most recent order for the same customer. The order found was the same one found in step 2.
5. T1 completed and was committed.

Isolation Test 9

This test demonstrates isolation for phantom protection between New-Order and Delivery transactions.

The execution of the above test proceeded as follows:

1. The NO_D_ID of all New_Order rows for a randomly selected warehouse and district was changed. The change was committed.
2. A delivery transaction T1 was started for the selected warehouse.
3. T1 was stopped immediately after reading the New_Order table for the selected warehouse and district. No qualifying row was found.
4. A New-Order transaction T2 was started for the same warehouse and district. T2 completed and was committed without being blocked by T1.
5. T1 was resumed and the New_Order rows for the table was read again. No qualifying row was found.
6. T1 completed and was committed.
7. The NO_D_ID of all New_Order rows for the selected warehouse and district was restored to the original value. The changes were committed.

Durability

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

Durable Media Failure

Durability from media failure was demonstrated on the 552 warehouse database. The standard driving mechanism was used to generate the transaction load of 5520 users for the Loss of Data.

Loss of Data/ Loss of Log

Loss of data was demonstrated on the 552 warehouse database. The standard driving mechanism was used to generate the transaction load of 5520 users for the test. To demonstrate recovery from a permanent failure of durable media containing TPC-C tables, the following steps were executed:

1. The 552 warehouse database was used for this test.
2. The database was backed up using SQL Server backup facilities.
3. A sum of D_NEXT_O_ID was taken.
4. 5520 users were logged in to the database and ran transactions.
5. The system was run at steady state for 5 minutes.
6. One disk drive in the transaction log array was removed with no effect on Windows 2003 or SQL Server.
7. One disk drive in the data array was removed causing SQL Server errors.
8. The RTE was allowed to continue running. Completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
9. The RTE was stopped.
10. SQL Server was stopped and restarted and a dump of the transaction log was taken.
11. SQL Server was stopped, Windows 2003 was shutdown and the machine powered off.
12. The failed disks were replaced.
13. The machine was powered up, Windows 2003 and SQL Server were started.
14. The TPC-C database was dropped and restored from backup.
15. The transaction log was restored and transactions rolled forward.
16. A new count of D_NEXT_O_ID was taken.
15. This number was compared with the number of new orders reported by the RTE. The difference was valid per the spec.

Instantaneous Interruption and Loss of Memory

Instantaneous Interruption and Loss of Memory were demonstrated on the database with 5512 warehouses in a single test. The standard driving mechanism was used to generate the transaction load of 55120 users for the test. To demonstrate recovery an instantaneous system interruption caused by powering off the Server, the following steps were executed:

1. The full database was used.
2. A sum of D_NEXT_O_ID was taken.
3. 55120 users were logged in to the database and ran transactions.
4. The system was run is steady state for 5 minutes
5. A checkpoint was executed and allowed to finish.
6. The system ran for an additional 30 seconds.
7. The Server was powered off by normal means, causing instantaneous interruption. No battery or UPS was providing power for the server.
8. The RTE was allowed to continue running. Completed transactions enroute from the clients were recorded. Error messages began appearing on the RTE screen.
9. The RTE was stopped.
10. The server was powered on again and rebooted.
11. SQL Server was restarted and automatically recovered.
12. A new count of D_NEXT_O_ID was taken.
13. This number was compared with the number of new orders reported by the RTE. The difference was valid per the spec.

Table Cardinality

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. (8.1.5.1)

The database was originally built with 5512 warehouses.

Table 3: Table Cardinality

Table	Cardinality as Benchmarked
Warehouse	5512
District	55120
Customer	165360000
History	165360000
NewOrder	49608000
Orders	165360000
OrderLine	165360000
Item	100000
Stock	551200000
Deleted Warehouses	0

Constant Values

The following values were used as constant value inputs to the NURand function for this benchmark.

Table 4: Constant Values

Function	Constant C Value
C_LAST (Build)	123
C_LAST (Run)	208

Data Distribution

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems. (8.1.5.2)

The Database was built using a total of 98 disks: 90 36GB for data, 8 146GB for log and OS and application software. The data drives were configured as hardware RAID 0. Logs and OS were configured as hardware RAID 10. 3 Dell Perc5e were configured with 1 logical drives each. Each logical drive spanned 30 disk drives. One internal PERC5i RAID Controller 0 was configured with 1 logical drive spanning 8 146GB drives. Each Windows 2003 data drive contained 3 partitions: partition 1 for customer/stock, partition 2 for miscellaneous, and partition 3 for backup. Partitions 1 and 2 were RAW file systems and partition 3 was formatted NTFS. The details are shown in Table 5.

Table 5: Data Distribution

W2K Disk Administration			Dell Perc5i					
Disk 0 544.49GB				Controller # 0				
Partition			Slot# **1**	Channels				
1	2	3	SAS ID	A				
C:	**B:**		0	**00-1**				
OS	**LOG**		1	**00-2**				
NTFS	**RAW**		2	**01-3**				
10.0GB	**534.50GB**		3	**01-4**				
			4	**02-5**				
			5	**02-6**				
			8	**03-7**				
			9	**03-8**				

W2K Disk Administration			3 Dell Perc5e SAS RAID Controllers						
Disk 1 1,001.24 GB **Disk 2 1,001.24 GB** **Disk 3 1,001.24 GB**			Controller HA-1-3						
Partition					Channels				
1	2	3	SAS ID	0	1	2	3	4	5
P:	**O:**		**0**	**00-1**	**01-1**	**02-1**	**03-1**	**04-1**	**05-1**
CS1	**MS1**		**1**	**00-2**	**01-2**	**02-2**	**03-2**	**04-2**	**05-2**
RAW	**RAW**		**2**	**00-3**	**01-3**	**02-3**	**03-3**	**04-3**	**05-3**
98.82 GB	**48.58 GB**		**3**	**00-4**	**01-4**	**02-4**	**03-4**	**04-4**	**05-4**
W:	**E:**	**R:**	**4**	**00-5**	**01-5**	**02-5**	**03-5**	**04-5**	**05-5**
CS2	**MS2**	**Backup1**	**5**	**00-6**	**01-6**	**02-6**	**03-6**	**04-6**	**05-6**
RAW	**RAW**	**NTFS**	**8**	**00-7**	**01-7**	**02-7**	**03-7**	**04-7**	**05-7**
119.99 GB	**58.98 GB**	**822.27 GB**	**9**	**00-8**	**01-8**	**02-8**	**03-8**	**04-8**	**05-8**
V:	**L:**	**T: Backup2**	**10**	**00-9**	**01-9**	**02-9**	**03-9**	**04-9**	**05-9**
CS3	**MS3**	**NTFS**	**11**	**00-10**	**01-10**	**02-10**	**03-10**	**04-10**	**05-10**
RAW	**RAW**	**853.84 GB**	**12**	**00-11**	**01-11**	**02-11**	**03-11**	**04-11**	**05-11**
98.82 GB	**48.58 GB**		**13**	**00-12**	**01-12**	**02-12**	**03-12**	**04-12**	**05-12**
			14	**00-13**	**01-13**	**02-13**	**03-13**	**04-13**	**05-13**
			15	**00-14**	**01-14**	**02-14**	**03-14**	**04-14**	**05-14**

***Comment**: Detailed diagrams for layout of database files on disks can widely vary, and it is difficult to provide exact guideline suitable for all implementations. The intent is to provide sufficient detail to allow independent reconstruction of the test database. The two figures below are examples of database layout descriptions and are not intended to depict or imply any optimal layout for the TPC-C database.*
8.1.5.3 A statement must be provided that describes:

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

Microsoft SQL Server Enterprise Edition is a relational DBMS.

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

Partition Mapping

The mapping of database partitions/replications must be explicitly described.
***Comment**: The intent is to provide sufficient detail about partitioning and replication to allow independent reconstruction of the test database. (8.1.5.3)*
An description of a database partitioning scheme is presented below as an example. The nomenclature of this example was outlined using the CUSTOMER table (in Clause 8.1.2.1), and has been extended to use the ORDER and ORDER_LINE tables as well.

The database was not replicated.

60 day Space Calculation

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). (8.1.5.5)

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

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

The results of the above steps yielded a requirement 344 GB (including mirror) to sustain the log for 8 hours. Space available on the transaction log volume was 534 GB (including mirror), indicating that enough storage was configured to sustain 8 hours of growth.

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

The details of the 60-day space requirement is shown in Appendix D.

Clause 5 -- Performance Metrics and Response Time Related Items

Measured TpmC

Measured tpmC must be reported. (8.1.6.1)

Measured TpmC	69,564
Price per TpmC	$.91

Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the Menu response time. (8.1.6.2)

Table 6: Transaction Response Times

Transaction	Average	90%	Maximum
New Order	0.19	0.26	5.00
Payment	0.13	0.15	1.66
Interactive Delivery	0.10	0.11	1.26
Stock Level	0.22	0.30	1.58
Order Status	0.18	0.24	4.68
Deferred Delivery	0.17	0.25	5.03
Menu	0.11	0.11	1.75

Think Times & Key Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type. (8.1.6.3)

Table 7: Transaction Key Times

Transaction	Minimum	Average	Maximum
New Order	18.02	18.03	19.17
Payment	3.02	3.03	4.18
Delivery	2.02	2.03	3.17
Stock Level	2.02	2.03	3.18
Order Status	2.02	2.03	3.16

Table 8: Transaction Think Times

Transaction	Minimum	Average	Maximum
New Order	0.00	12.05	120.44
Payment	0.00	12.05	120.43
Delivery	0.00	5.08	50.42
Stock Level	0.00	5.06	50.43
Order Status	0.00	10.05	100.43

Response Time Distribution Curves

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type. (8.1.6.4)

Figure 3: New Order Response Time Distribution

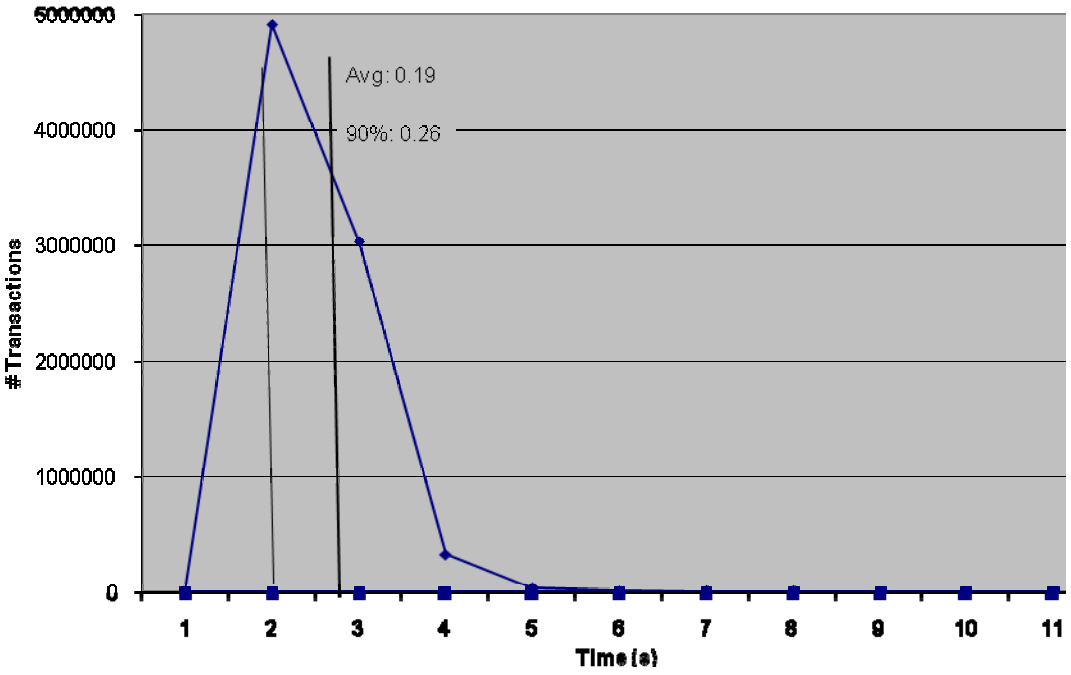


Figure 4: Payment Response Time Distribution

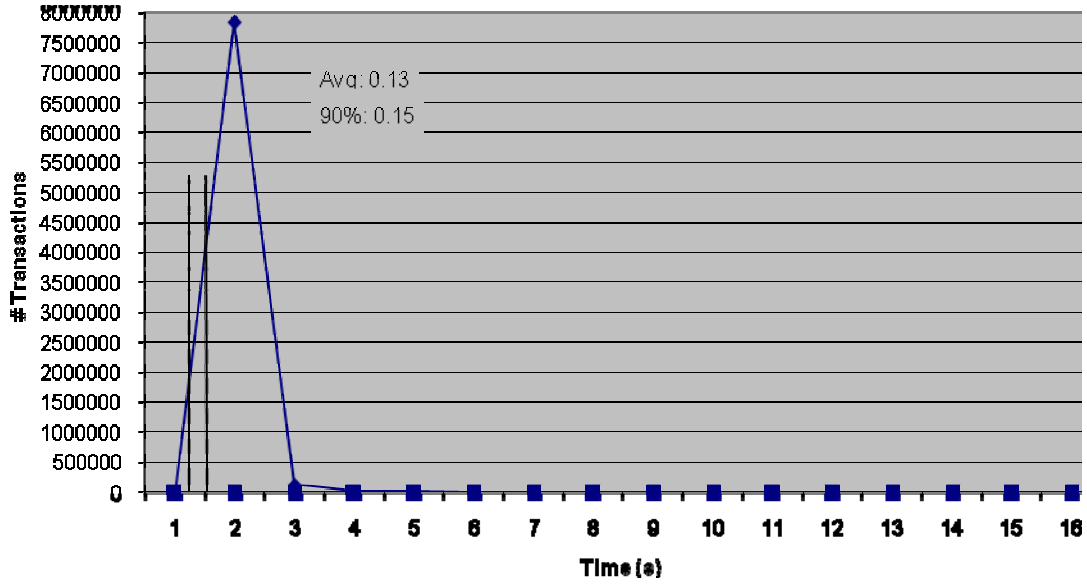


Figure 5: Order Status Response Time Distribution

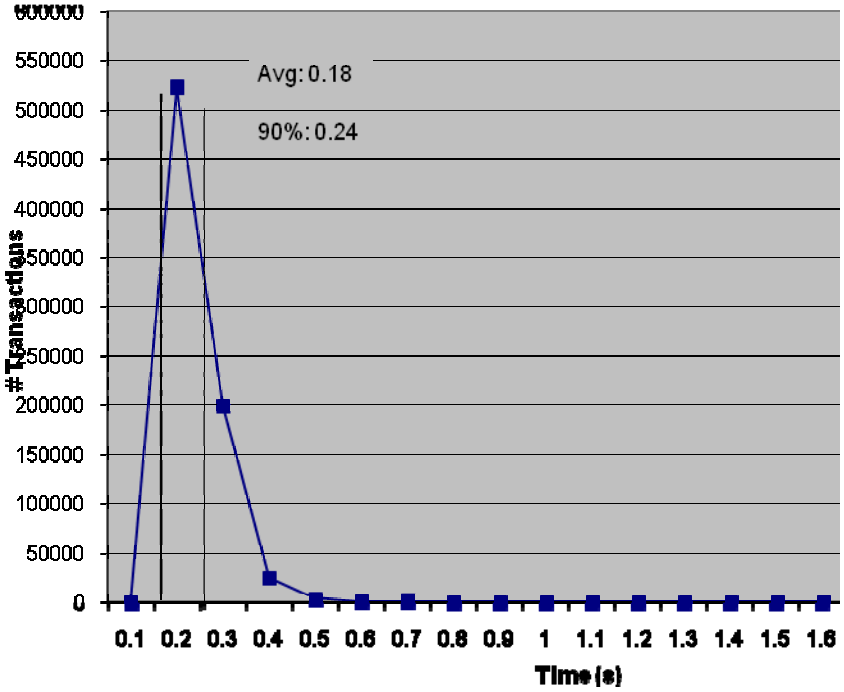


Figure 6: Delivery Response Time Distribution

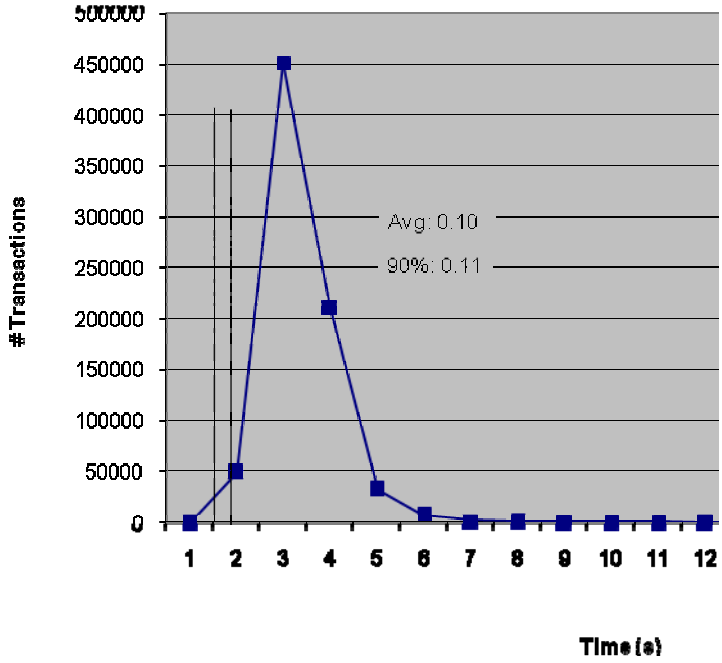
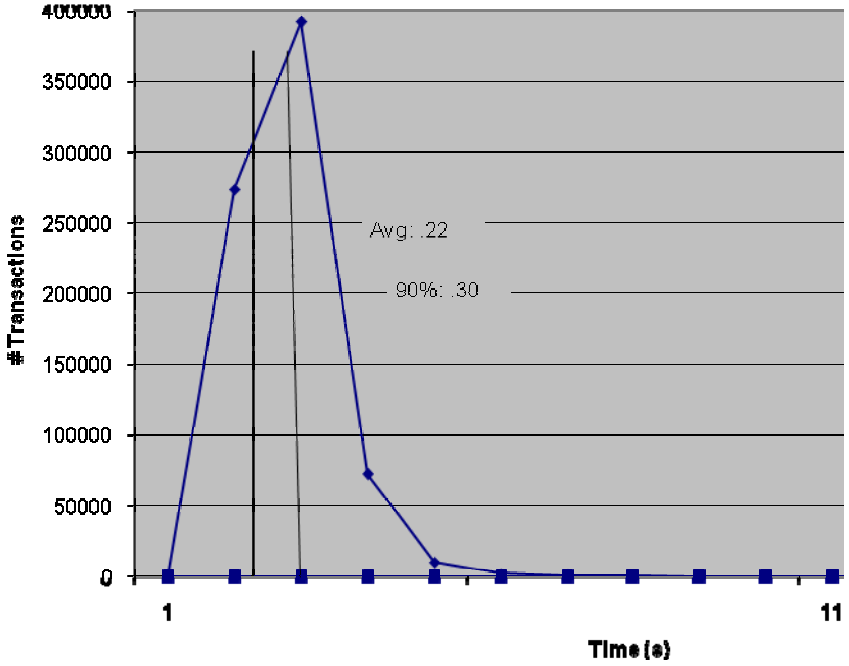


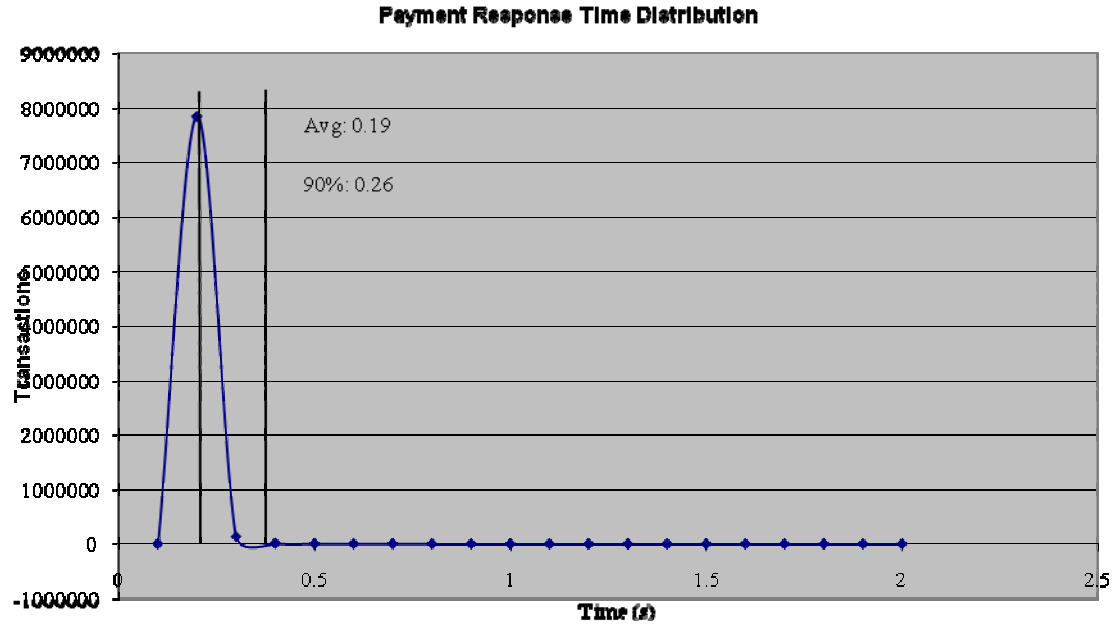
Figure 7: Stock Level Response Time Distribution



New-Order Response Time vs. Throughput Graph

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction. (8.1.6.5)

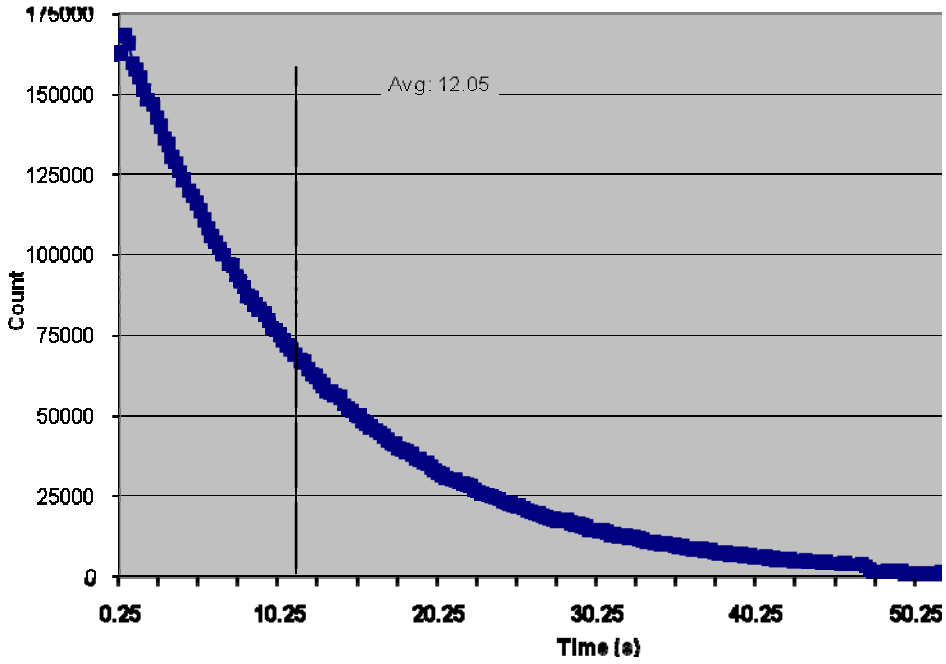
Figure 8: New Order Response Time vs. Throughput



New-Order Think Time Distribution Graph

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction (8.1.6.6)

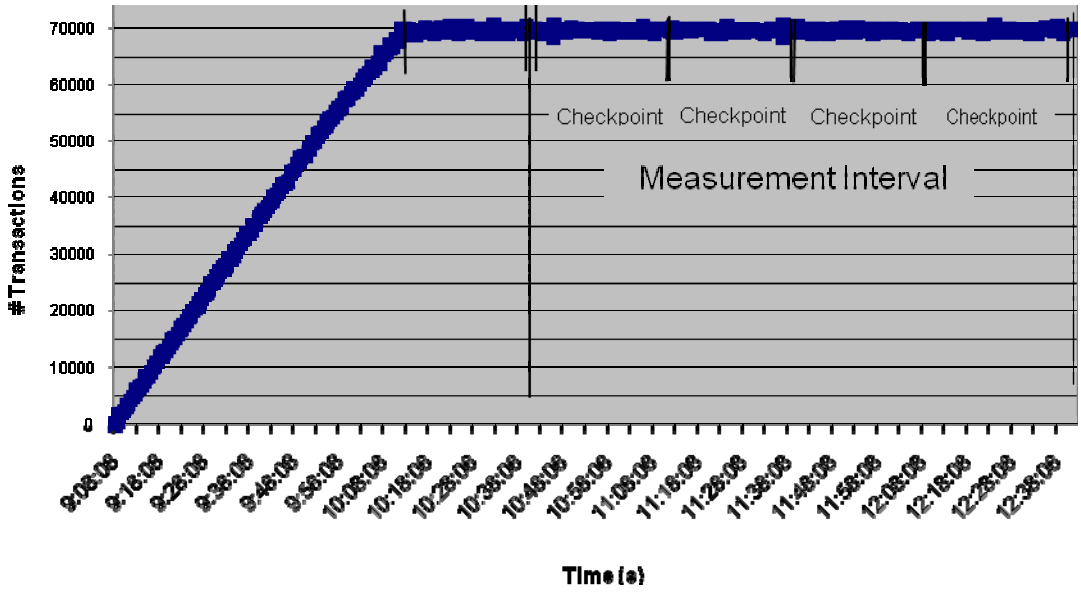
Figure 9: New Order Think Time Distribution



Steady-State Graph

A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction. (8.1.6.8)

Figure 10: New Order Throughput vs. Time



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. (8.1.6.9)

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

Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.), actually occurred during the measurement interval must be reported. (8.1.6.10)

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

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the HTTP request to the client. The transmission was timestamped. The return of the screen with the required response data was timestamped. The difference between these two timestamps was the response time for that transaction and was logged in the RTE log.

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

The RTE transmissions were sent to the web-based application program running on the client machines through Ethernet LANs. These web clients managed the emulated web browser interface as well as all requests to the database on the server. The applications communicated with the database server over another Ethernet LAN using the COM+ transaction monitor and Microsoft SQL Server DBLIB library and RPC calls.

To perform checkpoints at specific intervals, we set SQL Server *recovery interval* to the maximum allowable value and wrote a script to schedule multiple checkpoints at specific intervals. By setting the TRACE FLAG #3502, SQL Server logged the checkpoint beginning and ending time in the ERRORLOG file. The script included a wait time between each checkpoint equal to the measurement interval, which was 30 minutes. The checkpoint script was started manually after the RTE had all users logged in and sending transactions.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. Upon completion of the checkpoint, Microsoft SQL Server wrote a special record to the recovery log to indicate that all disk operations had been satisfied to this point.

Measurement Interval

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included. (8.1.6.12)

The measurement interval was 7200 seconds.

Measurement Period Duration and Checkpoint Duration

The start time and duration in seconds of at least the four (4) longest checkpoints during the measurement interval must be disclosed (see clause 5.5.2.2(2)) (8.1.6.11)
A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included. (8.1.6.12)

	Start	End	Duration
Measurement Interval	10:36:06	12:36:06	7,200
1st Checkpoint	10:47:32	11:05:26	1798
2nd Checkpoint	11:17:29	11:34:46	1798
3rd Checkpoint	11:47:27	12:04:55	1798
4th Checkpoint	12:17:25	12:35:00	1798

Transaction Mix

8.1.6.13 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 that was not adjusted during the run.

The percentage of the total mix for each transaction type must be disclosed. (8.1.6.14)

Table 9: Transaction Mix

Transaction	Percentage
New Order	44.82%
Payment	43.04%
Delivery	4.05%
Stock Level	4.05%
Order Status	4.04%

Other Metrics

The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. (8.1.6.15)
The average number of order-lines entered per New-Order transaction must be disclosed. (8.1.6.16)
The percentage of remote order-lines entered per New-Order transaction must be disclosed. (8.1.6.17)
The percentage of remote Payment transactions must be disclosed. (8.1.6.18)
The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. (8.1.6.19)
The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed. (8.1.6.20)

Table 10: Transaction Statistics

Transaction	Function	Value
New Order	Home Warehouse Items	99.85%
	Remote Warehouse Items	0.15%
	Rolled Back Transactions	1.00%
	Average Lines Per Order	10.00
Payment	Home Warehouse	85.00%
	Remote Warehouse	15.00%
	Non-Primary Key Access	60.00%
Order Status	Non-Primary Key Access	60.14%
Delivery	Skipped Transactions	0

RTE Parameters

The RTE input parameters, code fragments, functions, etc. used to generate each transaction input field must be disclosed. (8.1.7.1)
Comment: *The intent is to demonstrate the RTE was configured to generate transaction input data as specified in Clause 2.*

The RTE input parameters are listed in Appendix C - Tunable Parameters.

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. (8.1.7.2)

No components were emulated.

Benchmarked and Targeted System Configuration Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all 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). (8.1.7.3)

The driver system performed transaction data generation and communication to the client through the standard web browser (HTTP) protocol. It also captured and timestamped the SUT output data for post-processing of the reported metrics. No other functionality was included on the driver system.

Figures 1 & 2 of this report contain detailed diagrams of both the benchmark configuration and the priced configuration.

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). (8.1.7.4)

The network configurations of the benchmarked and priced configurations were identical.

Network Bandwidth

The bandwidth of the network(s) used in the tested/priced configuration must be disclosed. (8.1.7.5)

The bandwidth of the tested and priced networks were as follows:

- 1000 BaseT (1000 Mbit/sec) network segments between the RTE/Emulated Users and the Client.
- 1000 BaseT (1000 Mbit/sec) between the Client and Server.

Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed. (8.1.7.6)

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

Clause 7 -- Pricing Related Items

Hardware and Software List

A detailed list of hardware and software used in the priced system must be reported. Each separately orderable item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery 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) of price(s) must also be reported. (8.1.8.1)

The total 5-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. (8.1.8.2)

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

Availability 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 date for the priced system must be the date at which all components are committed to be available. (8.1.8.3)

Hardware Availability Date: March 9, 2007
Software Availability Date: March 9, 2007

Measured TpmC

A statement of the measured tpmC, as well as the respective calculations for 5-year pricing, price/performance (price/tpmC), and the availability date must be included. (8.1.8.4)

Maximum Qualified Throughput: 69,564 tpmC
Price Performance Metric: $.91

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. (8.1.8.5)

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

Usage Pricing

For any usage pricing, the sponsor must disclose (8.1.8.6):

- *Usage level at which the component was priced.*
- *A statement of the company policy allowing such pricing.*

***Comment**: Usage pricing may include, but is not limited to, the operating system and database management software.*

The component pricing based on usage is shown below:

- 1 Microsoft Windows Server 2003, Standard x64 Edition License.
- 1 Microsoft Windows Server 2003, Standard Edition License.
- 1 Microsoft SQL Server 2005 Standard x64 Edition License (1 processor).
- 1 Microsoft Visual C++ Standard Edition.
- 3 Year Support for Hardware Components.

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). Clause 6.1 describes the Server and Client components. An example of the standard pricing sheet is shown in Appendix B. (8.1.8.7)
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. See example in Appendix B. (8.1.8.8)

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

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. (8.1.9.1)
A review of the pricing model is required to ensure that all components required are priced (see Clause 9.2.8). The auditor is not required to review the final Full Disclosure Report or the final pricing prior to issuing the attestations letter. (8.1.9.2)

This TPC-C benchmark has been audited by Lorna Livingtree of Performance Metrics.

Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
c/o Adminstrator, TPC
Presidio of San Francisco
Bldg 572B Ruger St.
San Francisco, CA 94129-0920
Phone: (415) 561-6272, Fax (415)561-6120
www.tpc.org

or:

Dell Inc.
One Dell Way
Round Rock, TX 78682
Attention: Mike Molloy, Ph.D.

March 9, 2007

Mr. Dan Hambrick
Dell Computer Corporation
One Dell Way
Round Rock, TX 78682

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

Platform: Dell PowerEdge 2900
Database Manager: Microsoft SQL Server 2005 x64 Standard Edition
Operating System: Microsoft Windows Server 2003 Standard x64 Edition
Transaction Monitor: COM+

System Under Test: Dell PowerEdge 2900 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
1 quad core Intel @ 2.3 Ghz	Main: 24 GB	90 @36GB 8 @ 73GB	0.26	69,564

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

* The transactions were correctly implemented.
* The database files were properly sized.
* The database was properly scaled with 5,512 warehouses, all of which were active during the measured interval.
* The ACID properties were successfully demonstrated.
* Data loss durability was demonstrated on a subset of the SUT configured with a database properly populated for 552 warehouses.
* Input data was generated according to the specified percentages.
* Eight hours of mirrored log space was present on the tested system.
* The data for the 60 days space calculation was verified.
* The steady state portion of the test was 120 minutes.
* One checkpoint was taken in steady state before the measured interval opened.
* Four checkpoints were completed inside the measured interval.
* The system pricing was checked for major components and maintenance.
* Third party quotes were verified for compliance.

Auditor Notes:
None

Sincerely,

Lorna Livingtree
Auditor

Appendix A - Application Source Code

tpcc.dll ISAPI DLL Source Code

isapi_dll/src/tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

Isapi_dll/src/tpcc.h

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

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

#define TP_MAX_RETRIES                                                              50


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

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


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

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

          CTPCC_BASE          *pTxn;

} CLIENTDATA, *PCLIENTDATA;


//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
          int                           iNumEntries;                                                    //total allocated
terminal array entries
          int                           iFreeList;                                                              //next
available terminal array element or -1 if none
```

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

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


        enum WEBERROR
        {
                NO_ERR,
                ERR_COMMAND_UNDEFINED,
                ERR_D_ID_INVALID,
                ERR_DELIVERY_CARRIER_ID_RANGE,
                ERR_DELIVERY_CARRIER_INVALID,
                ERR_DELIVERY_MISSING_OCD_KEY,
                ERR_DELIVERY_THREAD_FAILED,
                ERR_GETPROCADDR_FAILED,
                ERR_HTML_ILL_FORMED,
                ERR_INVALID_SYNC_CONNECTION,
                ERR_INVALID_TERMID,
                ERR_LOADDLL_FAILED,
                ERR_MAX_CONNECTIONS_EXCEEDED,
                ERR_MEM_ALLOC_FAILED,
                ERR_MISSING_REGISTRY_ENTRIES,
                ERR_NEWORDER_CUSTOMER_INVALID,
                ERR_NEWORDER_CUSTOMER_KEY,
                ERR_NEWORDER_DISTRICT_INVALID,
                ERR_NEWORDER_FORM_MISSING_DID,
                ERR_NEWORDER_ITEMID_INVALID,
                ERR_NEWORDER_ITEMID_RANGE,
                ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
                ERR_NEWORDER_MISSING_IID_KEY,
                ERR_NEWORDER_MISSING_QTY_KEY,
                ERR_NEWORDER_MISSING_SUPPW_KEY,
                ERR_NEWORDER_NOITEMS_ENTERED,
                ERR_NEWORDER_QTY_INVALID,
                ERR_NEWORDER_QTY_RANGE,
                ERR_NEWORDER_QTY_WITHOUT_SUPPW,
                ERR_NEWORDER_SUPPW_INVALID,
                ERR_NO_SERVER_SPECIFIED,
                ERR_ORDERSTATUS_CID_AND_CLT,
                ERR_ORDERSTATUS_CID_INVALID,
                ERR_ORDERSTATUS_CLT_RANGE,
                ERR_ORDERSTATUS_DID_INVALID,
                ERR_ORDERSTATUS_MISSING_CID_CLT,
                ERR_ORDERSTATUS_MISSING_CID_KEY,
                ERR_ORDERSTATUS_MISSING_CLT_KEY,
                ERR_ORDERSTATUS_MISSING_DID_KEY,
                ERR_PAYMENT_CDI_INVALID,
                ERR_PAYMENT_CID_AND_CLT,
                ERR_PAYMENT_CUSTOMER_INVALID,
                ERR_PAYMENT_CWI_INVALID,
                ERR_PAYMENT_DISTRICT_INVALID,
                ERR_PAYMENT_HAM_INVALID,
                ERR_PAYMENT_HAM_RANGE,
                ERR_PAYMENT_LAST_NAME_TO_LONG,
                ERR_PAYMENT_MISSING_CDI_KEY,
                ERR_PAYMENT_MISSING_CID_CLT,
                ERR_PAYMENT_MISSING_CID_KEY,
                ERR_PAYMENT_MISSING_CLT,
                ERR_PAYMENT_MISSING_CLT_KEY,
                ERR_PAYMENT_MISSING_CWI_KEY,
                ERR_PAYMENT_MISSING_DID_KEY,
                ERR_PAYMENT_MISSING_HAM_KEY,
                ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
                ERR_STOCKLEVEL_THRESHOLD_INVALID,
                ERR_STOCKLEVEL_THRESHOLD_RANGE,
                ERR_VERSION_MISMATCH,
                ERR_W_ID_INVALID
        };

class CWEBCLNT_ERR : public CBaseErr
{
        public:

                CWEBCLNT_ERR(WEBERROR Err)
                {
                        m_Error = Err;
                        m_szTextDetail = NULL;
                        m_SystemErr = 0;
                        m_szErrorText = NULL;
                };

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

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

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

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

};

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

//function prototypes

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

isapi_dll/src/tpcc.rc

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

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

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

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

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

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

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

#endif    // !_MAC


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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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


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

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
```

```
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 179
        TOPMARGIN, 7
        BOTTOMMARGIN, 88
    END
END
#endif    // APSTUDIO_INVOKED

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


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


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

isapi_dll/src/tpcc.cpp

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

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

#include <sqltypes.h>

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

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

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

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

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

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


#define   LEN_ERR_STRING      256

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

```
char               szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

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

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

static    CRITICAL_SECTION             TermCriticalSection;

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

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

// For deferred Delivery txns:

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

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

// configuration settings from registry
TPCCREGISTRYDATA             Reg;


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

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

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

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

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

          try
          {
                    switch( ul_reason_for_call )
                    {
                              case DLL_PROCESS_ATTACH:
                                        {
                                                  DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
                                                  GetComputerName(szMyComputerName, &dwSize);
                                                  szMyComputerName[dwSize] = 0;
                                        }

                                        DisableThreadLibraryCalls((HMODULE)hModule);
                                        InitializeCriticalSection(&TermCriticalSection);

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

```
                                        dwDelBuffSize = min( Reg.dwMaxPendingDeliveries, 10000 );  // min with 10000 as
a sanity constraint
                                        dwNumDeliveryThreads = min( Reg.dwNumberOfDeliveryThreads, 100 );  // min with
100 as a sanity constraint

                                        TermInit();

                                        // load DLL for txn monitor
                                        if (Reg.eTxnMon == TUXEDO)
                                        {
                                                strcpy( szDllName, Reg.szPath );
                                                strcat( szDllName, "tpcc_tuxedo.dll");
                                                hLibInstanceTm = LoadLibrary( szDllName );
                                                if (hLibInstanceTm == NULL)
                                                        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

                                                // get function pointer to wrapper for class constructor
                                                pCTPCC_TUXEDO_new = (TYPE_CTPCC_TUXEDO*)
GetProcAddress(hLibInstanceTm,"CTPCC_TUXEDO_new");
                                                if (pCTPCC_TUXEDO_new == NULL)
                                                        throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                        }
                                        else if (Reg.eTxnMon == ENCINA)
                                        {
                                                strcpy( szDllName, Reg.szPath );
                                                strcat( szDllName, "tpcc_encina.dll");
                                                hLibInstanceTm = LoadLibrary( szDllName );
                                                if (hLibInstanceTm == NULL)
                                                        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );
                                                // get function pointer to wrapper for class constructor
                                                pCTPCC_ENCINA_new = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_new");
                                                pCTPCC_ENCINA_post_init = (TYPE_CTPCC_ENCINA*)
GetProcAddress(hLibInstanceTm,"CTPCC_ENCINA_post_init");
                                                if (pCTPCC_ENCINA_new == NULL)
                                                        throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                                        }
                                        else if (Reg.eTxnMon == COM)
                                        {
                                                strcpy( szDllName, Reg.szPath );
                                                strcat( szDllName, "tpcc_com.dll");
                                                hLibInstanceTm = LoadLibrary( szDllName );
                                                if (hLibInstanceTm == NULL)
                                                        throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

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

                                        // load DLL for database connection
                                        if ((Reg.eTxnMon == None) || (dwNumDeliveryThreads > 0))
                                        {
                                                if (Reg.eDB_Protocol == DBLIB)
                                                {
                                                        strcpy( szDllName, Reg.szPath );
                                                        strcat( szDllName, "tpcc_dblib.dll");
                                                        hLibInstanceDb = LoadLibrary( szDllName );
                                                        if (hLibInstanceDb == NULL)
                                                                throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                                        // get function pointer to wrapper for class constructor
                                                        pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                                        if (pCTPCC_DBLIB_new == NULL)
                                                                throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
                                                }
                                                else if (Reg.eDB_Protocol == ODBC)
                                                {
                                                        strcpy( szDllName, Reg.szPath );
                                                        strcat( szDllName, "tpcc_odbc.dll");
                                                        hLibInstanceDb = LoadLibrary( szDllName );
                                                        if (hLibInstanceDb == NULL)
                                                                throw new CWEBCLNT_ERR( ERR_LOADDLL_FAILED,
szDllName, GetLastError() );

                                                        // get function pointer to wrapper for class constructor
                                                        pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                                        if (pCTPCC_ODBC_new == NULL)
                                                                throw new CWEBCLNT_ERR( ERR_GETPROCADDR_FAILED,
szDllName, GetLastError() );
                                                }
```

```
                                        }

                                        if (dwNumDeliveryThreads)
                                        {
                                                // for deferred delivery txns:
                                                hDoneEvent = CreateEvent( NULL, TRUE /* manual reset */, FALSE /*
initially not signalled */, NULL );
                                                InitializeCriticalSection(&DelBuffCriticalSection);
                                                hWorkerSemaphore = CreateSemaphore( NULL, 0, dwDelBuffSize, NULL );
                                                dwDelBuffFreeCount = dwDelBuffSize;

                                                InitJulianTime(NULL);

                                                // create unique log file name based on delilog-yymmdd-hhmm.log
                                                SYSTEMTIME Time;
                                                GetLocalTime( &Time );
                                                wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-%2.2d%2.2d.log",
                                                                Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute );
                                                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);
                                        }

                                        DeleteCriticalSection(&TermCriticalSection);

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

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

                                        Sleep(500);
                                        break;

                                default:
                                        /* nothing */;
                        }
                }
                catch (CBaseErr *e)
                {
                        WriteMessageToEventLog( e->ErrorText() );
                        delete e;
                        TerminateExtension(0);
                        return FALSE;
                }
                catch (...)
                {
                        WriteMessageToEventLog(TEXT("Unhandled exception.  DLL could not load."));
                        TerminateExtension(0);
                        return FALSE;
```

```
          }

          return TRUE;
}


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

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

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

          return TRUE;
}

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

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

          TermDeleteAll();
          return TRUE;
}


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

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

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


#ifdef ICECAP
          StartCAP();
#endif

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

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

```
                                        char szTmp[128];
                                        wsprintf( szTmp, "Invalid term ID; TermId = %d", TermId );
                                        WriteMessageToEventLog( szTmp );

                                        throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
                                }

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

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


                        switch(iCmd)
                        {
                        case 0:
                                WelcomeForm(pECB, szBuffer);
                                break;
                        case 1:
                                switch( FormId )
                                {
                                        case WELCOME_FORM:
                                        case MAIN_MENU_FORM:
                                                break;
                                        case NEW_ORDER_FORM:
                                                ProcessNewOrderForm(pECB, TermId, szBuffer);
                                                break;
                                        case PAYMENT_FORM:
                                                ProcessPaymentForm(pECB, TermId, szBuffer);
                                                break;
                                        case DELIVERY_FORM:
                                                ProcessDeliveryForm(pECB, TermId, szBuffer);
                                                break;
                                        case ORDER_STATUS_FORM:
                                                ProcessOrderStatusForm(pECB, TermId, szBuffer);
                                                break;
                                        case STOCK_LEVEL_FORM:
                                                ProcessStockLevelForm(pECB, TermId, szBuffer);
                                                break;
                                }
                                break;
                        case 2:
                                // new-order selected from menu; display new-order input form
                                MakeNewOrderForm(TermId, NULL, INPUT_FORM, szBuffer);
                                break;
                        case 3:
                                // payment selected from menu; display payment input form
                                MakePaymentForm(TermId, NULL, INPUT_FORM, szBuffer);
                                break;
                        case 4:
                                // delivery selected from menu; display delivery input form
                                MakeDeliveryForm(TermId, NULL, INPUT_FORM, szBuffer);
                                break;
                        case 5:
                                // order-status selected from menu; display order-status input form
                                MakeOrderStatusForm(TermId, NULL, INPUT_FORM, szBuffer);
                                break;
                        case 6:
                                // stock-level selected from menu; display stock-level input form
                                MakeStockLevelForm(TermId, NULL, INPUT_FORM, szBuffer);
                                break;
                        case 7:
                                // ExitCmd
                                TermDelete(TermId);
                                WelcomeForm(pECB, szBuffer);
                                break;
                        case 8:
                                SubmitCmd(pECB, szBuffer);
                                break;
                        case 9:
                                // menu
                                MakeMainMenuForm(TermId, Term.pClientData[TermId].iSyncId, szBuffer);
                                break;
                        case 10:
                                // CMD=Clear
                                // resets all connections; should only be used when no other connections are active
                                TermDeleteAll();
                                TermInit();
                                WelcomeForm(pECB, szBuffer);
                                break;
                        case 11:  // CMD=Stats
                                StatsCmd(pECB, szBuffer);
                                break;
                        }
                }
                catch (CBaseErr *e)
                {
                        ErrorForm( pECB, e->ErrorType(), e->ErrorNum(), TermId, iSyncId, e->ErrorText(), szBuffer );
                        delete e;
                }
```

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

#ifdef ICECAP
          StopCAP();
#endif


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

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

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


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

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

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

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}


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

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

          DELIVERY_TRANSACTION                    delivery;
          PDELIVERY_DATA                                  pDeliveryData;
          TXN_RECORD_TPCC_DELIV_DEF     txnDeliRec;

          DWORD                                           index;
          HANDLE                                          handles[2];

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

          assert(txnDelilog != NULL);

          try
          {
                    if (Reg.eDB_Protocol == ODBC)
                              pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, szMyComputerName,
Reg.szDbName, Reg.szSPPrefix );
                    else if (Reg.eDB_Protocol == DBLIB)
                              pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                    pDeliveryData = pTxn->BuffAddr_Delivery();
```

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

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

				ZeroMemory(&txnDeliRec, sizeof(txnDeliRec));
				txnDeliRec.TxnType = TXN_REC_TYPE_TPCC_DELIV_DEF;

				// make a local copy of current entry from delivery buffer and increment buffer
index
				EnterCriticalSection(&DelBuffCriticalSection);
				delivery = *(pDelBuff+dwDelBuffBusyIndex);
				dwDelBuffFreeCount++;
				dwDelBuffBusyIndex++;
				if (dwDelBuffBusyIndex == dwDelBuffSize)	// wrap-around if at end of
buffer
					dwDelBuffBusyIndex = 0;

				LeaveCriticalSection(&DelBuffCriticalSection);

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

				txnDeliRec.w_id = pDeliveryData->w_id;
				txnDeliRec.o_carrier_id = pDeliveryData->o_carrier_id;
				txnDeliRec.TxnStartT0 = Get(x64)Time(&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)(Get(x64)Time(&trans_end) - txnDeliRec.TxnStartT0);
				txnDeliRec.DeltaTxnExec = (int)(Get(x64)Time(&trans_end) -
Get(x64)Time(&trans_start));

				if (txnDelilog != NULL)
					txnDelilog->WriteToLog(&txnDeliRec);
			}
		}
		catch (CBaseErr *e)
		{
			char szTmp[1024];
			wsprintf( szTmp, "Error in Delivery Txn thread.  %s", 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:
	delete pTxn;
	_endthread();
}
```

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

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

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

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

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

          return bError;
}

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

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

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

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

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

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

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

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

/* FUNCTION: void WelcomeForm
```

```
 *
 */

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

          //welcome to tpc-c html form buffer, this is first form client sees.
          strcpy( szBuffer,   "<HTML><HEAD><TITLE>TPC-C Web Client</TITLE></HEAD><BODY>"
                                                    "<B><BIG>Microsoft TPC-C Web Client (ver 4.20)</BIG></B>
<BR> <BR>"
                                                    "<font face=\"Courier New\"><PRE>"
                                                    "Compiled: "__DATE__", "__TIME__" <BR>"
                                                    "Source:   "__FILE__" ("__TIMESTAMP__") <BR>"
                                                    "</PRE></font>"
                                                    "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"0\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"1\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"TERMID\" VALUE=\"0\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"0\">"
                                                    "<INPUT TYPE=\"hidden\" NAME=\"VERSION\" VALUE=\""
WEBCLIENT_VERSION "\">"
                              );

          sprintf( szTmp,     "Configuration Settings: <BR><font face=\"Courier New\" color=\"blue\"><PRE>"
                                                  "Txn Monitor            = <B>%s</B><BR>"
                                                  "Database protocol      = <B>%s</B><BR>"
                                                  "Max Connections        = <B>%d</B><BR>"
                                                  "# of Delivery Threads  = <B>%d</B><BR>"
                                                  "Max Pending Deliveries = <B>%d</B><BR>"
                              , szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
                              Reg.dwMaxConnections, dwNumDeliveryThreads, dwDelBuffSize );
          strcat( szBuffer, szTmp);

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

          if (Reg.eTxnMon == None)
                    // connection options may be specified when not using a txn monitor
                    sprintf( szTmp,     "Please enter your database options for this connection:<BR>"
                                                    "<font face=\"Courier New\" color=\"blue\"><PRE>"
                                                    "DB Server    = <INPUT NAME=\"db_server\" SIZE=20
VALUE=\"%s\"><BR>"
                                                    "DB User ID   = <INPUT NAME=\"db_user\" SIZE=20
VALUE=\"%s\"><BR>"
                                                    "DB Password  = <INPUT NAME=\"db_passwd\" SIZE=20
VALUE=\"%s\"><BR>"
                                                    "DB Name      = <INPUT NAME=\"db_name\" SIZE=20
VALUE=\"%s\"><BR>"
                                                    "</PRE></font>"
                                       , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
          else
                    // if using a txn monitor, connection options are determined from registry; can't
                    // set per user.  show options fyi
                    sprintf( szTmp,     "Database options which will be used by the transaction monitor:<BR>"
                                                    "<font face=\"Courier New\" color=\"blue\"><PRE>"
                                                    "DB Server              = <B>%s</B><BR>"
                                                    "DB User ID             = <B>%s</B><BR>"
                                                    "DB Password            = <B>%s</B><BR>"
                                                    "DB Name                = <B>%s</B><BR>"
                                                    "</PRE></font>"
                                       , Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword, Reg.szDbName );
          strcat( szBuffer, szTmp);

          sprintf( szTmp,     "Please enter your Warehouse and District for this session:<BR>"
                                                  "<font face=\"Courier New\" color=\"blue\"><PRE>" );
          strcat( szBuffer, szTmp);
          strcat( szBuffer,   "Warehouse ID = <INPUT NAME=\"w_id\" SIZE=6><BR>"
                                                    "District ID  = <INPUT NAME=\"d_id\" SIZE=2><BR>"
                                                    "</PRE></font><HR>"
                                                    "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Submit\">"
                                                    "</FORM></BODY></HTML>");
}

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

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

          char      szVersion[32]       = { 0 };
          char      szServer[32]        = { 0 };
```

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

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

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

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

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


          iNewTerm = TermAdd();

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

          try
          {
                    if (Reg.eTxnMon == TUXEDO)
                              Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new();
                    else if (Reg.eTxnMon == ENCINA)
                              Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new();
                    else if (Reg.eTxnMon == COM)
                              Term.pClientData[iNewTerm].pTxn = pCTPCC_COM_new( Reg.bCOM_SinglePool );
                    else if (Reg.eDB_Protocol == ODBC)
                              Term.pClientData[iNewTerm].pTxn = pCTPCC_ODBC_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase, Reg.szSPPrefix );
                    else if (Reg.eDB_Protocol == DBLIB)
                              Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new( szServer, szUser, szPassword,
szMyComputerName, szDatabase );
          }
          catch (...)
          {
                    TermDelete(iNewTerm);
                    throw;              // pass exception upward
          }

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

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

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

          EnterCriticalSection(&TermCriticalSection);

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

          LeaveCriticalSection(&TermCriticalSection);

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

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

```
        {
                {       ERR_COMMAND_UNDEFINED,                          "Command
undefined."
        },
                {       ERR_D_ID_INVALID,                               "Invalid District
ID Must be 1 to 10."                                                    },
                {       ERR_DELIVERY_CARRIER_ID_RANGE,          "Delivery Carrier ID out of
range must be 1 - 10."                                  },
                {       ERR_DELIVERY_CARRIER_INVALID,           "Delivery Carrier ID invalid must be
numeric 1 - 10."                                                },
                {       ERR_DELIVERY_MISSING_OCD_KEY,           "Delivery missing Carrier ID key
\"OCD*\"."                                                      },
                {       ERR_DELIVERY_THREAD_FAILED,             "Could not start delivery
worker thread."                                                 },
                {       ERR_GETPROCADDR_FAILED,                         "Could not map proc
in DLL.  GetProcAddr error.  DLL="                      },
                {       ERR_HTML_ILL_FORMED,                            "Required key field
is missing from HTML string."                           },
                {       ERR_INVALID_SYNC_CONNECTION,            "Invalid Terminal Sync ID."
                                                                        },
                {       ERR_INVALID_TERMID,                             "Invalid Terminal
ID."
        },
                {       ERR_LOADDLL_FAILED,                             "Load of DLL
failed.  DLL="
        },
                {       ERR_MAX_CONNECTIONS_EXCEEDED,           "No connections available.  Max
Connections is probably too low."   },
                {       ERR_MISSING_REGISTRY_ENTRIES,           "Required registry entries are missing.
Rerun INSTALL to correct."  },
                {       ERR_NEWORDER_CUSTOMER_INVALID,                  "New Order customer id
invalid data type, range = 1 to 3000."          },
                {       ERR_NEWORDER_CUSTOMER_KEY,                      "New Order missing Customer
key \"CID*\"."                                                  },
                {       ERR_NEWORDER_DISTRICT_INVALID,          "New Order District ID
Invalid range 1 - 10."                                  },
                {       ERR_NEWORDER_FORM_MISSING_DID,                  "New Order missing District
key \"DID*\"."                                                  },
                {       ERR_NEWORDER_ITEMID_INVALID,            "New Order Item Id is wrong data type,
must be numeric."                                               },
                {       ERR_NEWORDER_ITEMID_RANGE,                      "New Order Item Id is out of
range. Range = 1 to 999999."                    },
                {       ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,      "New Order Item_Id field entered
without a corresponding Supp_W."        },
                {       ERR_NEWORDER_MISSING_IID_KEY,           "New Order missing Item Id key
\"IID*\"."                                                      },
                {       ERR_NEWORDER_MISSING_QTY_KEY,           "New Order Missing Qty key \"Qty##*\"."
                                                                        },
                {       ERR_NEWORDER_MISSING_SUPPW_KEY,                 "New Order missing Supp_W key
\"SP##*\"."                                                     },
                {       ERR_NEWORDER_NOITEMS_ENTERED,           "New Order No order lines entered."
                                                                        },
                {       ERR_NEWORDER_QTY_INVALID,                       "New Order Qty invalid must
be numeric range 1 - 99."                               },
                {       ERR_NEWORDER_QTY_RANGE,                                 "New Order Qty is
out of range. Range = 1 to 99."                                 },
                {       ERR_NEWORDER_QTY_WITHOUT_SUPPW,         "New Order Qty field entered
without a corresponding Supp_W."                },
                {       ERR_NEWORDER_SUPPW_INVALID,                     "New Order Supp_W invalid
data type must be numeric."                             },
                {       ERR_NO_SERVER_SPECIFIED,                        "No Server name specified."
                                                                        },
                {       ERR_ORDERSTATUS_CID_AND_CLT,            "Order Status Only Customer ID or Last
Name may be entered, not both."         },
                {       ERR_ORDERSTATUS_CID_INVALID,            "Order Status Customer ID invalid,
range must be numeric 1 - 3000."        },
                {       ERR_ORDERSTATUS_CLT_RANGE,                      "Order Status Customer last
name longer than 16 characters."                },
                {       ERR_ORDERSTATUS_DID_INVALID,            "Order Status District invalid, value
must be numeric 1 - 10."                },
                {       ERR_ORDERSTATUS_MISSING_CID_CLT,        "Order Status Either Customer ID or
Last Name must be entered."             },
                {       ERR_ORDERSTATUS_MISSING_CID_KEY,        "Order Status missing Customer key
\"CID*\"."                                              },
                {       ERR_ORDERSTATUS_MISSING_CLT_KEY,        "Order Status missing Customer Last
Name key \"CLT*\"."                             },
                {       ERR_ORDERSTATUS_MISSING_DID_KEY,        "Order Status missing District key
\"DID*\"."                                              },
                {       ERR_PAYMENT_CDI_INVALID,                        "Payment Customer district
invalid must be numeric."                               },
                {       ERR_PAYMENT_CID_AND_CLT,                        "Payment Only Customer ID or
Last Name may be entered, not both."    },
                {       ERR_PAYMENT_CUSTOMER_INVALID,           "Payment Customer data type invalid,
must be numeric."                               },
                {       ERR_PAYMENT_CWI_INVALID,                        "Payment Customer Warehouse
invalid, must be numeric."                              },
                {       ERR_PAYMENT_DISTRICT_INVALID,           "Payment District ID is invalid, must
be 1 - 10."                                             },
                {       ERR_PAYMENT_HAM_INVALID,                        "Payment Amount invalid data
type must be numeric."                          },
                {       ERR_PAYMENT_HAM_RANGE,                          "Payment Amount out
of range, 0 - 9999.98."                                 },
```

```
                    {         ERR_PAYMENT_LAST_NAME_TO_LONG,                          "Payment Customer last name
longer than 16 characters."                                                   },
                    {         ERR_PAYMENT_MISSING_CDI_KEY,                  "Payment missing Customer district key
\"CDI*\"."                                                                    },
                    {         ERR_PAYMENT_MISSING_CID_CLT,                  "Payment Either Customer ID or Last
Name must be entered."                                                        },
                    {         ERR_PAYMENT_MISSING_CID_KEY,                  "Payment missing Customer Key
\"CID*\"."                                                                    },
                    {         ERR_PAYMENT_MISSING_CLT_KEY,                  "Payment missing Customer Last Name key
\"CLT*\"."                                                                    },
                    {         ERR_PAYMENT_MISSING_CWI_KEY,                  "Payment missing Customer Warehouse key
\"CWI*\"."                                                                    },
                    {         ERR_PAYMENT_MISSING_DID_KEY,                  "Payment missing District Key
\"DID*\"."                                                                    },
                    {         ERR_PAYMENT_MISSING_HAM_KEY,                  "Payment missing Amount key \"HAM*\"."
                                                                          },
                    {         ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,   "Stock Level; missing Threshold key \"TT*\"."
                                                                          },
                    {         ERR_STOCKLEVEL_THRESHOLD_INVALID,             "Stock Level; Threshold value must be
in the range = 1 - 99."                                                   },
                    {         ERR_STOCKLEVEL_THRESHOLD_RANGE,                         "Stock Level Threshold out of
range, range must be 1 - 99."                                             },
                    {         ERR_VERSION_MISMATCH,                                             "Invalid version
field.  RTE and Web Client are probably out of sync." },
                    {         ERR_W_ID_INVALID,                                                 "Invalid Warehouse
ID."
          },
                    {         0,
          ""
                                                                          }
          };

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

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

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

}

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

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

          if ( !(ptr=strstr(*pQueryString, pKey)) )
                    goto ErrorExit;
```

```
          ptr += strlen(pKey);
          if ( *ptr != '=' )
                    goto ErrorExit;
          ptr++;

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

          *pQueryString = ptr;
          return;

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

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

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

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

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

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

          *pQueryString = ptr;
          return atoi(ptr0);

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

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

void TermInit(void)
```

```
{
          EnterCriticalSection(&TermCriticalSection);

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

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

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

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

          LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

          LeaveCriticalSection(&TermCriticalSection);
}

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

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

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

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

```
                    // do not bump existing connection; instead, return error to requestor.
                    if ((GetTickCount() - iTickCount) < 60000)
                    {
                              LeaveCriticalSection(&TermCriticalSection);
                              throw new CWEBCLNT_ERR( ERR_MAX_CONNECTIONS_EXCEEDED );
                    }
          }

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

          LeaveCriticalSection(&TermCriticalSection);
          return iNewTerm;
}

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

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

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

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

                    LeaveCriticalSection(&TermCriticalSection);
          }
}


/* FUNCTION: MakeErrorForm
 */

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

/* FUNCTION: MakeMainMenuForm
 */

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

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

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

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

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

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

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

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

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

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

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

                    strcpy( szForm+c,
                              "District: <INPUT NAME=\"DID*\" SIZE=1>                        Date:<BR>"
                              "Customer:  <INPUT NAME=\"CID*\" SIZE=4>   Name:                    Credit:
%Disc:<BR>"
                              "Order Number:           Number of Lines:           W_tax:         D_tax:<BR> <BR>"
                              " Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price    Amount<BR>"
                              "  <INPUT NAME=\"SP00*\" SIZE=4>  <INPUT NAME=\"IID00*\" SIZE=6>
<INPUT NAME=\"Qty00*\" SIZE=1><BR>"
```

```
                            "  <INPUT NAME=\"SP01*\" SIZE=4>  <INPUT NAME=\"IID01*\" SIZE=6>
<INPUT NAME=\"Qty01*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP02*\" SIZE=4>  <INPUT NAME=\"IID02*\" SIZE=6>
<INPUT NAME=\"Qty02*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP03*\" SIZE=4>  <INPUT NAME=\"IID03*\" SIZE=6>
<INPUT NAME=\"Qty03*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP04*\" SIZE=4>  <INPUT NAME=\"IID04*\" SIZE=6>
<INPUT NAME=\"Qty04*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP05*\" SIZE=4>  <INPUT NAME=\"IID05*\" SIZE=6>
<INPUT NAME=\"Qty05*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP06*\" SIZE=4>  <INPUT NAME=\"IID06*\" SIZE=6>
<INPUT NAME=\"Qty06*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP07*\" SIZE=4>  <INPUT NAME=\"IID07*\" SIZE=6>
<INPUT NAME=\"Qty07*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP08*\" SIZE=4>  <INPUT NAME=\"IID08*\" SIZE=6>
<INPUT NAME=\"Qty08*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP09*\" SIZE=4>  <INPUT NAME=\"IID09*\" SIZE=6>
<INPUT NAME=\"Qty09*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP10*\" SIZE=4>  <INPUT NAME=\"IID10*\" SIZE=6>
<INPUT NAME=\"Qty10*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP11*\" SIZE=4>  <INPUT NAME=\"IID11*\" SIZE=6>
<INPUT NAME=\"Qty11*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP12*\" SIZE=4>  <INPUT NAME=\"IID12*\" SIZE=6>
<INPUT NAME=\"Qty12*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP13*\" SIZE=4>  <INPUT NAME=\"IID13*\" SIZE=6>
<INPUT NAME=\"Qty13*\" SIZE=1><BR>"
                            "  <INPUT NAME=\"SP14*\" SIZE=4>  <INPUT NAME=\"IID14*\" SIZE=6>
<INPUT NAME=\"Qty14*\" SIZE=1><BR>"
                            "Execution Status:                                                  Total:<BR>"
                            "</font></PRE><HR>"
                            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
                            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                            "</FORM></HTML>"
                            );
        }
        else
        {
                c += wsprintf(szForm+c, "Warehouse: %6.6d   District: %2.2d                       Date: ",
                            pNewOrderData->w_id,
                            pNewOrderData->d_id);

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

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

                if ( bValid )
                {
                        c += sprintf(szForm+c,
                                                "%%Disc: %5.2f           <BR>"
                                                "Order Number: %8.8d  Number of Lines: %2.2d        W_tax:
%5.2f   D_tax: %5.2f  <BR> <BR>"
                                                " Supp_W  Item_Id  Item Name                 Qty  Stock
B/G  Price    Amount<BR>",
                                    100.0*pNewOrderData->c_discount,
                                    pNewOrderData->o_id,
                                    pNewOrderData->o_ol_cnt,
                                    100.0 * pNewOrderData->w_tax,
                                    100.0 * pNewOrderData->d_tax);

                                for(i=0; i<pNewOrderData->o_ol_cnt; i++)
                                {
                                        c += sprintf(szForm+c, "%6.6d   %6.6d   %-24s  %2.2d    %3.3d    %1.1s   $%6.2f
$%7.2f  <BR>",
                                                pNewOrderData->OL[i].ol_supply_w_id,
                                                pNewOrderData->OL[i].ol_i_id,
                                                pNewOrderData->OL[i].ol_i_name,
                                                pNewOrderData->OL[i].ol_quantity,
                                                pNewOrderData->OL[i].ol_stock,
                                                pNewOrderData->OL[i].ol_brand_generic,
                                                pNewOrderData->OL[i].ol_i_price,
                                                pNewOrderData->OL[i].ol_amount );
                                }
                }
                else
                {
                        c += wsprintf(szForm+c,
                                    "%Disc:<BR>"
                                    "Order Number: %8.8d  Number of Lines:              W_tax:         D_tax:<BR>
<BR>"
                                    " Supp_W  Item_Id  Item Name                 Qty  Stock  B/G  Price
Amount<BR>"
                                    , pNewOrderData->o_id);
```

```
                        i = 0;
                }

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

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

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

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

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

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

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

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

                        , Term.pClientData[iTermId].w_id, pPaymentData->d_id
```

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

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

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

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

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

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

}

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

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

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

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

```
                                "District: %2.2d<BR>"
                                "Customer: %4.4d   Name: %-16s %-2s %-16s<BR>",
                                pOrderStatusData->d_id, pOrderStatusData->c_id,
                                pOrderStatusData->c_first, pOrderStatusData->c_middle, pOrderStatusData->c_last);

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

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

                        for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
                        {
                                c += sprintf(szForm+c, "  %6.6d     %6.6d     %2.2d     $%8.2f      %2.2d-%2.2d-
%4.4d<BR>",
                                        pOrderStatusData->OL[i].ol_supply_w_id,
                                        pOrderStatusData->OL[i].ol_i_id,
                                        pOrderStatusData->OL[i].ol_quantity,
                                        pOrderStatusData->OL[i].ol_amount,
                                        pOrderStatusData->OL[i].ol_delivery_d.day,
                                        pOrderStatusData->OL[i].ol_delivery_d.month,
                                        pOrderStatusData->OL[i].ol_delivery_d.year);
                        }

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

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

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

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

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

        if ( bInput )
        {
                strcpy( szForm+c,
                        "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR> <BR>"
                        "Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR> <BR> <BR> <BR> </font></PRE><HR>"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                        "</BODY></FORM></HTML>" );
        }
        else
        {
                wsprintf( szForm+c,
                        "Carrier Number: %2.2d<BR> <BR>"
                        "Execution Status: %s <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
                        " <BR> <BR> <BR> <BR> <BR> <BR> <BR> </font></PRE>"
                        "<HR><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..NewOrder..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-Status..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-Level..\">"
                        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/* FUNCTION: ProcessDeliveryForm
 *
 * PURPOSE:         This function gets and validates the input data from the delivery form
 *                          filling in the required input variables. It then calls the PostDeliveryInfo
```

```
 *                          Api, The client is then informed that the transaction has been posted.
 *
 * ARGUMENTS:       EXTENSION_CONTROL_BLOCK       *pECB     passed in structure pointer from inetsrv.
 *                                  int                                                         iTermId   client
browser terminal id
 *
 */

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

          PDELIVERY_DATA      pDelivery;

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

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

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

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

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

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

          PSTOCK_LEVEL_DATA   pStockLevel;

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

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

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

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

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

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

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

          static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
```

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

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

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

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

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

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

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

          pNewOrderData->o_ol_cnt = items;
}

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

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

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

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

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

          if ( bCustIdBlank )
          {         // customer id is blank, so last name must be entered
                    GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp), ERR_PAYMENT_MISSING_CLT_KEY);
```

```
                    if ( szTmp[0] == 0 )
                              throw new CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

                    _strupr( szTmp );
                    if ( strlen(pPaymentData->c_last) > LAST_NAME_LEN )
                              throw new CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );
                    strcpy(pPaymentData->c_last, szTmp);
          }
          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;

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

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

                    _strupr( szTmp );
                    if ( strlen(pOrderStatusData->c_last) > LAST_NAME_LEN )
                              throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );
                    strcpy(pOrderStatusData->c_last, szTmp);
          }
          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;
}
```

isapi_dll/src/resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by tpcc.rc
//
#define IDD_DIALOG1                     101

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        102
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1000
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

common/src/ReadRegistry.cpp

```
/*        FILE:               READREGISTRY.CPP
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                    not yet audited
 *
 *        PURPOSE:  Implementation for TPC-C Tuxedo class.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.20.000 - first version
 */


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

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

```
                return TRUE;

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

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

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

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

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

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

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

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

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

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

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

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

        RegCloseKey(hKey);

        return FALSE;
}
```

common/src/ReadRegistry.h

```
/*      FILE:           ReadRegistry.h
 *                              Microsoft TPC-C Kit Ver. 4.20.000
 *                              Copyright Microsoft, 1999
 *           All Rights Reserved
 *
 *                              not audited
 *
```

```
 *        PURPOSE:  Header for registry related code.
 *
 *  Change history:
 *                  4.20.000 - first version
 */

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

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

//This structure defines the data necessary to keep distinct for each terminal or client connection.
typedef struct _TPCCREGISTRYDATA
{
          enum DBPROTOCOL eDB_Protocol;
          enum TXNMON eTxnMon;
          BOOL bCOM_SinglePool;
          DWORD dwMaxConnections;
          DWORD dwMaxPendingDeliveries;
          DWORD dwNumberOfDeliveryThreads;
          char szPath[128];
          char szDbServer[32];
          char szDbName[32];
          char szDbUser[32];
          char szDbPassword[32];
          wchar_t szSPPrefix[32];       //tpcc_odbc.dll stored procedures prefix
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

common/src/error.h

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

#pragma once

#ifndef _INC_STRING
          #include <string.h>
#endif

const int m_szMsg_size = 512;
const int m_szApp_size = 64;
const int m_szLoc_size = 64;

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
          int                iError;                              //error id of message
          char      szMsg[256];                           //message to sent to browser
} SERRORMSG;

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


#define ERR_TYPE_LOGIC                                                                  -1                //logic
error in program; internal error
#define ERR_SUCCESS                                                                     0                 //success
(a non-error error)
#define ERR_BAD_ITEM_ID                                                                 1
          //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST                                                2                 //expected delivery
post failed
#define   ERR_TYPE_WEBDLL                                                               3                 //tpcc
web generated error
#define   ERR_TYPE_SQL                                                                  4                 //sql
server generated error
#define   ERR_TYPE_DBLIB                                                                5                 //dblib
generated error
#define   ERR_TYPE_ODBC                                                                 6                 //odbc
generated error
```

```
#define ERR_TYPE_SOCKET                                                         7                   //error
on communication socket client rte only
#define ERR_TYPE_DEADLOCK                                               8                   //dblib and odbc
only deadlock condition
#define ERR_TYPE_COM                                                            9                   //error
from COM call
#define ERR_TYPE_TUXEDO                                                         10                  //tuxedo
error
#define ERR_TYPE_OS                                                             11
          //operating system error
#define ERR_TYPE_MEMORY                                                         12                  //memory
allocation error
#define ERR_TYPE_TPCC_ODBC                                              13                  //error from tpcc
odbc txn module
#define ERR_TYPE_TPCC_DBLIB                                             14                  //error from tpcc
dblib txn module
#define ERR_TYPE_DELISRV                                                15                  //delivery server
error
#define ERR_TYPE_TXNLOG                                                         16                  //txn log
error
#define ERR_TYPE_BCCONN                                                         17
          //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN                                              18                  //Benchcraft
connection class
#define ERR_TYPE_ENCINA                                                         19                  //Encina
error
#define ERR_TYPE_COMPONENT                                              20                  //error from COM
component
#define ERR_TYPE_RTE                                                            21
          //Benchcraft rte
#define ERR_TYPE_AUTOMATION                                             22                  //Benchcraft
automation errors
#define ERR_TYPE_DRIVER                                                         23                  //Driver
engine errors
#define ERR_TYPE_RTE_BASE                                               24                  //Framework errors
#define   ERR_BUF_OVERFLOW                                              25                  //Buffer overflow
during receive
#define ERR_TYPE_SOAP_HTTP                                              26                  //HTTP/SOAP dll
generated error
// 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

#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:

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


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

                    m_szApp             = new char[m_szApp_size];
                    GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
          }



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


                    if (szLoc)
                    {
```

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

                m_szApp             = new char[m_szApp_size];
                GetModuleFileName(GetModuleHandle(NULL), m_szApp, m_szApp_size);
        }

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

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

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

                j += wsprintf(szTmp+j, "%s\n", ErrorText());

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

        char *GetApp(void) { return m_szApp; }
        char *GetLocation(void) { return m_szLoc; }
        virtual int ErrorNum() { return m_idMsg; }



        virtual int ErrorType() = 0;  // a value which distinguishes the kind of error that occurred
        virtual char *ErrorText() = 0; // a string (i.e., human readable) representation of 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,
                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 *ErrorText(void);

};
```

```
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,
          };

                              CSystemErr(Action eAction, LPCTSTR szLocation);
                              CSystemErr(int iError, Action eAction, LPCTSTR szLocation);
          int                 ErrorType() { return ERR_TYPE_OS;};
          char      *ErrorText(void);
          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 *ErrorText() {return ERR_INS_MEMORY;}
};


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

          int ErrorType() {return ERR_BUF_OVERFLOW;}
          char *ErrorText() {return ERR_INS_BUF_OVERFLOW;}
};
```

common/src/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 dblib, so redefined here.  Note: we are using the symbol "__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has been declared.
#ifndef __SQLTYPES
          typedef struct
          {
                        short                         /* SQLSMALLINT */   year;
                        unsigned short      /* SQLUSMALLINT */  month;
                        unsigned short      /* SQLUSMALLINT */  day;
                        unsigned short      /* SQLUSMALLINT */  hour;
                        unsigned short      /* SQLUSMALLINT */  minute;
                        unsigned short      /* SQLUSMALLINT */  second;
                        unsigned long       /* SQLUINTEGER */   fraction;
          } TIMESTAMP_STRUCT;
#endif

// possible values for exec_status_code after transaction completes
enum EXEC_STATUS
{
          eOK,                          // 0      "Transaction commited."
          eInvalidItem,       // 1      "Item number is not valid."
          eDeliveryFailed     // 2      "Delivery Post Failed."
};

// transaction structures
typedef struct
{
          // input params
          long                                  ol_supply_w_id;
          long                                  ol_i_id;
          short                                 ol_quantity;

          // output params
          char                                  ol_i_name[I_NAME_LEN+1];
          char                                  ol_brand_generic[BRAND_LEN+1];
          double                                ol_i_price;
          double                                ol_amount;
          short                                 ol_stock;
} OL_NEW_ORDER_DATA;

typedef struct
{
          // input params
          long                w_id;
          short               d_id;
          long                c_id;
          short               o_ol_cnt;

          // output params
          EXEC_STATUS                           exec_status_code;
          char                c_last[LAST_NAME_LEN+1];
          char                c_credit[CREDIT_LEN+1];
          double              c_discount;
          double              w_tax;
          double              d_tax;
          long                          o_id;
          short                                 o_commit_flag;
          TIMESTAMP_STRUCT    o_entry_d;
          short               o_all_local;
          double              total_amount;
```

```
        OL_NEW_ORDER_DATA   OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

typedef struct
{       // input params
        long                                    w_id;
        short                                   d_id;
        long                                    c_id;
        short                                   c_d_id;
        long                                    c_w_id;
        double                                  h_amount;
        char                            c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                             exec_status_code;
        TIMESTAMP_STRUCT     h_date;
        char                                    w_street_1[ADDRESS_LEN+1];
        char                                    w_street_2[ADDRESS_LEN+1];
        char                                    w_city[ADDRESS_LEN+1];
        char                                    w_state[STATE_LEN+1];
        char                                    w_zip[ZIP_LEN+1];
        char                                    d_street_1[ADDRESS_LEN+1];
        char                                    d_street_2[ADDRESS_LEN+1];
        char                                    d_city[ADDRESS_LEN+1];
        char                                    d_state[STATE_LEN+1];
        char                                d_zip[ZIP_LEN+1];
        char                                    c_first[FIRST_NAME_LEN+1];
        char                                    c_middle[MIDDLE_NAME_LEN + 1];
        char                                    c_street_1[ADDRESS_LEN+1];
        char                                    c_street_2[ADDRESS_LEN+1];
        char                                    c_city[ADDRESS_LEN+1];
        char                                    c_state[STATE_LEN+1];
        char                                    c_zip[ZIP_LEN+1];
        char                                    c_phone[PHONE_LEN+1];
        TIMESTAMP_STRUCT     c_since;
        char                                    c_credit[CREDIT_LEN+1];
        double                                  c_credit_lim;
        double                                  c_discount;
        double                                  c_balance;
        char                                    c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

typedef struct
{
        long                                ol_i_id;
        long                        ol_supply_w_id;
        short                               ol_quantity;
        double                              ol_amount;
        TIMESTAMP_STRUCT    ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{       // input params
        long                w_id;
        short               d_id;
        long                c_id;
        char                c_last[LAST_NAME_LEN+1];

        // output params
        EXEC_STATUS                          exec_status_code;
        char                c_first[FIRST_NAME_LEN+1];
        char                c_middle[MIDDLE_NAME_LEN+1];
        double              c_balance;
        long                o_id;
        TIMESTAMP_STRUCT    o_entry_d;
        short               o_carrier_id;
        OL_ORDER_STATUS_DATA OL[MAX_OL_ORDER_STATUS_ITEMS];
        short               o_ol_cnt;
} ORDER_STATUS_DATA, *PORDER_STATUS_DATA;


typedef struct
{       // input params
        long                        w_id;
        short                       o_carrier_id;

        // output params
        EXEC_STATUS                         exec_status_code;
        SYSTEMTIME                  queue_time;
        long                                o_id[10];          // id's of delivered orders for districts 1 to 10
} DELIVERY_DATA, *PDELIVERY_DATA;

//This structure is used for posting delivery transactions and for writing them to the delivery server.
typedef struct _DELIVERY_TRANSACTION
{
        SYSTEMTIME          queue;                      //time delivery transaction queued
        long                w_id;                       //delivery warehouse
        short               o_carrier_id;       //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{       // input params
        long                                    w_id;
```

```
          short                                   d_id;
          short                                   threshold;

          // output params
          EXEC_STATUS                             exec_status_code;
          long                                    low_stock;
} STOCK_LEVEL_DATA, *PSTOCK_LEVEL_DATA;
```

common/src/txn_base.h

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

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class DllDecl CTPCC_BASE
{
          public:
                    CTPCC_BASE(void) {};
                    virtual ~CTPCC_BASE(void) {};

                    virtual PNEW_ORDER_DATA                 BuffAddr_NewOrder()           = 0;
                    virtual PPAYMENT_DATA                   BuffAddr_Payment()            = 0;
                    virtual PDELIVERY_DATA                  BuffAddr_Delivery()           = 0;
                    virtual PSTOCK_LEVEL_DATA     BuffAddr_StockLevel()         = 0;
                    virtual PORDER_STATUS_DATA    BuffAddr_OrderStatus()        = 0;

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

install\src\install.c

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

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

#include "resource.h"

#define   WM_INITTEXT                   WM_USER+100

HICON               hIcon;
HINSTANCE hInst;
```

```
DWORD                       versionExeMS;
DWORD                       versionExeLS;
DWORD                       versionExeMM;
DWORD                       versionDllMS;
DWORD                       versionDllLS;

// TPC-C registry settings
TPCCREGISTRYDATA    Reg;

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

static    int               iIISMajorVersion;
static    int               iNumberOfProcessors;

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


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

SYSTEM_INFO siSysInfo;

BOOL install_com(char *szDllPath);

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

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

          hInst = hInstance;

          InitCommonControls();

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

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

          DestroyIcon(hIcon);
          return 0;
}

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

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

```
                                hRes = LoadResource(hInst, hResInfo );
                                pSrc = (BYTE *)LockResource(hRes);
                                pDst = (unsigned char *)malloc(dwSize+1);
                                if ( pDst )
                                {
                                        memcpy(pDst, pSrc, dwSize);
                                        pDst[dwSize] = 0;
                                        SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pDst);
                                        free(pDst);
                                }
                                else
                                        SetDlgItemText(hwnd, IDC_LICENSE, (const char *)pSrc);
                                return TRUE;
                        case WM_DESTROY:
                                DeleteObject(hFont);
                                return TRUE;
                        case WM_COMMAND:
                                if ( wParam == IDOK )
                                        EndDialog(hwnd, TRUE);
                                if ( wParam == IDCANCEL )
                                        EndDialog(hwnd, FALSE);
                        default:
                                break;
                }
                return FALSE;
}

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
                switch(uMsg)
                {
                        case WM_INITDIALOG:
                                switch(lParam)
                                {
                                        case 1:
                                        case 2:
                                                SetDlgItemText(hwnd, IDC_RESULTS, "TPC-C Web Client Installed");
                                                break;
                                }
                                return TRUE;
                        case WM_COMMAND:
                                if ( wParam == IDOK )
                                        EndDialog(hwnd, TRUE);
                                break;
                        default:
                                break;
                }
                return FALSE;
}

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

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

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

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

                                // set default values
                                ZeroMemory( &Reg, sizeof(Reg) );
                                Reg.dwNumberOfDeliveryThreads = 4;
                                Reg.dwMaxConnections = 100;
                                Reg.dwMaxPendingDeliveries    = 100;
                                Reg.eDB_Protocol = DBLIB;
                                Reg.eTxnMon = None;
                                strcpy(Reg.szDbServer,                  "");
                                strcpy(Reg.szDbName,                    "tpcc");
                                strcpy(Reg.szDbUser,                    "sa");
                                strcpy(Reg.szDbPassword,        "");
```

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

		ReadTPCCRegistrySettings( &Reg );
		ReadRegistrySettings();

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

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

		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_THEAD_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);

		CheckDlgButton(hwnd, IDC_DBLIB, 0);
		CheckDlgButton(hwnd, IDC_ODBC,  0);
		if ( Reg.eDB_Protocol == DBLIB )
			CheckDlgButton(hwnd, IDC_DBLIB, 1);
		else
			CheckDlgButton(hwnd, IDC_ODBC,  1);

		// 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_TUXEDO, 0);
		CheckDlgButton(hwnd, IDC_TM_MTS,    0);
		CheckDlgButton(hwnd, IDC_TM_ENCINA,     0);
		switch (Reg.eTxnMon)
		{
		case None:
			CheckDlgButton(hwnd, IDC_TM_NONE,   1);
			break;
		case TUXEDO:
			CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
			break;
		case ENCINA:
			CheckDlgButton(hwnd, IDC_TM_ENCINA, 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 IDC_DBLIB:
					return TRUE;
				case IDC_ODBC:
					return TRUE;
				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    szErrTxt[128];

        // 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_DBLIB) )
        {
                Reg.eDB_Protocol = DBLIB;
                rc = 1;
        }
        else if ( IsDlgButtonChecked(hwnd, IDC_ODBC) )
        {
                Reg.eDB_Protocol = ODBC;
                rc = 2;
        }

        if ( IsDlgButtonChecked(hwnd, IDC_TM_NONE) )
                Reg.eTxnMon = None;
        else if ( IsDlgButtonChecked(hwnd, IDC_TM_TUXEDO) )
                Reg.eTxnMon = TUXEDO;
        else if ( IsDlgButtonChecked(hwnd, IDC_TM_MTS) )
                Reg.eTxnMon = COM;
        else if ( IsDlgButtonChecked(hwnd, IDC_TM_ENCINA) )
                Reg.eTxnMon = ENCINA;

        iPoolThreadLimit = GetDlgItemInt(hwnd, ED_IIS_MAX_THEAD_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);

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

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

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

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

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

	// register com proxy stub
	strcpy(szFullName, szDllPath);
	strcat(szFullName, "tpcc_com_ps.dll");
	if (!RegisterDLL(szFullName))
	{
		ShowWindow(hwnd, SW_SHOWNA);
		DestroyWindow(hDlg);
		strcpy( szErrTxt, "Error occured when registering " );
		strcat( szErrTxt, szFullName );
		MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
		EndDialog(hwnd, 0);
		return;
	}

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

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

	Sleep(100);

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

	EndDialog(hwnd, rc);
	return;
}

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

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

	if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\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
			// with the MaxPoolThreads rather thatn PoolThreadLimit
			// for ease of coding, we are just going to stuff the value into iPoolThreadLimit
  size = sizeof(iPoolThreadLimit);
			if ( RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type, (char *)&iPoolThreadLimit, &size)
== ERROR_SUCCESS )
			if ( !iPoolThreadLimit )
				iPoolThreadLimit =  iMaxPhysicalMemory * 2;
		}
```

```
                    else
                    {
                              size = sizeof(iPoolThreadLimit);
                              if ( RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type, (char *)&iPoolThreadLimit, &size)
== ERROR_SUCCESS )
                              if ( !iPoolThreadLimit )
                                        iPoolThreadLimit =  iMaxPhysicalMemory * 2;
                    }

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

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

                    RegCloseKey(hKey);
          }

          if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\W3SVC\\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 proivided in the dialog and then set
                    // PoolThreadLimit to MaxPoolThreads * number of processors on this system
                    if ( iIISMajorVersion == 6 )
                    {
                              iMaxPoolThreads = iPoolThreadLimit;
                              iPoolThreadLimit = iMaxPoolThreads * iNumberOfProcessors;
                              RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char *)&iPoolThreadLimit,
sizeof(iPoolThreadLimit));
                              RegSetValueEx(hKey, "MaxPoolThreads", 0, REG_DWORD, (char *)&iMaxPoolThreads,
sizeof(iMaxPoolThreads));
                    }
                    else
                    {
                              RegSetValueEx(hKey, "PoolThreadLimit", 0, REG_DWORD, (char *)&iPoolThreadLimit,
sizeof(iPoolThreadLimit));
                    }

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

                    RegFlushKey(hKey);
                    RegCloseKey(hKey);
          }

          if ( (iRc=RegCreateKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\W3SVC\\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, 16));
                    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 MSVCR70.DLL
	strcpy( szLastFileName, "msvcr70.dll" );
	if (!FileFromResource( "MSVCRT70", IDR_MSVCRT701, szWindowsPath, szLastFileName ))
		return 0;
	SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	UpdateDialog(hDlg);

	// install tpcc_dblib.dll
	strcpy( szLastFileName, "tpcc_dblib.dll" );
	if (!FileFromResource( "DBLIB_DLL", IDR_DBLIB_DLL, szDllPath, 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 tuxapp.exe
	strcpy( szLastFileName, "tuxapp.exe" );
	if (!FileFromResource( "TUXEDO_APP", IDR_TUXEDO_APP, szDllPath, szLastFileName ))
		return 0;
	//SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
	//UpdateDialog(hDlg);

	// install tpcc_tuxedo.dll
	strcpy( szLastFileName, "tpcc_tuxedo.dll" );
	if (!FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_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;

        versionDllMS = 0;
        versionDllLS = 0;
        if ( _access(szDLLPath, 00) == 0 )
        {
                dwSize = GetFileVersionInfoSize(szDLLPath, &d);
                if ( dwSize )
                {
                        ptr = (char *)malloc(dwSize);
                        GetFileVersionInfo(szDLLPath, 0, dwSize, ptr);
                        VerQueryValue(ptr, "\\",&vs, &dwBytes);
                        versionDllMS = vs->dwProductVersionMS;
                        versionDllLS = 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      szErrTxt[128];
          FILE      *fErrorFile;

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

          irc = system("IIS6_CONFIG.CMD");

          // since the return code from the command file is always 1,
          // check to see if the file iis6_config.err exists
          // if it does, then something hosed
```

```
          fErrorFile = fopen("IIS6_CONFIG.err","r");
          if ( fErrorFile != NULL )
          {
                    ShowWindow(hwnd, SW_SHOWNA);
                    DestroyWindow(hDlg);
                    strcpy( szErrTxt, "IIS6 configuration error." );
                    strcat( szErrTxt, "Check iis6_config.err" );
                    MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
                    EndDialog(hwnd, 0);
                    return;
          }
}
```

install\src\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                               bstrTemp, bstrTemp2, bstrTemp3, bstrTemp4;
          _bstr_t                               bstrDllPath = szDllPath;
          _variant_t                            vTmp, vKey;
          long                                  lActProp, lCount, lCountCo, lCountItf, lCountMethod;
          bool                                  bTmp;


          CoInitializeEx(NULL, COINIT_MULTITHREADED);

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

          if (!SUCCEEDED(hr)) goto Error;

          bstrTemp = "Applications";

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

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

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

          // iterate through applications to delete existing "TPC-C" application (if any)
          while (lCount > 0)
```

```
{
          hr = pCatalogCollectionApp->get_Item(lCount - 1, (IDispatch**) &pCatalogObjectApp);
          if (!SUCCEEDED(hr)) goto Error;

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

          if (wcscmp(vTmp.bstrVal, L"TPC-C"))
          {
                    lCount--;
                    continue;
          }
          else
          {
                    hr = pCatalogCollectionApp->Remove(lCount - 1);
                    if (!SUCCEEDED(hr)) goto Error;
                    break;
          }
}

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

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

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

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

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

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

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

pCatalogObjectApp->Release();
pCatalogObjectApp = NULL;

bstrTemp  = "TPC-C";                                    // app name
bstrTemp2 =         bstrDllPath + "tpcc_com_all.dll";               // DLL
bstrTemp3 =         bstrDllPath + "tpcc_com_all.tlb";               // type library (TLB)
bstrTemp4 =         bstrDllPath + "tpcc_com_ps.dll";                // proxy/stub dll

hr = pCOMAdminCat->InstallComponent(bstrTemp,

bstrTemp2,

bstrTemp3,

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


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

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

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

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

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

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

                    bstrTemp = "ConstructorString";
                    bstrTemp2 = "dummy string (do not remove)";
                    vTmp = bstrTemp2;
                    hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
                    if (!SUCCEEDED(hr)) goto Error;

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

                    bstrTemp = "MaxPoolSize";
                    vTmp.Clear();       // clear variant so it isn't stored as a bool (_variant_t feature)
                    vTmp = (long)30;
                    hr = pCatalogObjectCo->put_Value(bstrTemp, vTmp);
                    if (!SUCCEEDED(hr)) goto Error;

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



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

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

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

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

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

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

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

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

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

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

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

                                        pCatalogObjectMethod->Release();
                                        pCatalogObjectMethod = NULL;

                                        lCountMethod--;
                              }

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

                              pCatalogObjectItf->Release();
                              pCatalogObjectItf = NULL;
```

```
                        lCountItf--;

                }


                pCatalogObjectCo->Release();
                pCatalogObjectCo = NULL;

                lCountCo--;
        }


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

        pCatalogCollectionApp->Release();
        pCatalogCollectionApp = NULL;

        pCatalogCollectionCo->Release();
        pCatalogCollectionCo = NULL;

        pCatalogCollectionItf->Release();
        pCatalogCollectionItf = NULL;

        pCatalogCollectionMethod->Release();
        pCatalogCollectionMethod = NULL;


Error:
        CoUninitialize();

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

db_dblib_dll/src/tpcc_dblib.cpp

```
/*        FILE:               TPCC_DBLIB.CPP
 *                                      Microsoft TPC-C Kit Ver. 4.42.000
 *                                      Copyright Microsoft, 2002
 *             All Rights Reserved
 *
 *                                      Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *        PURPOSE:  Implements dblib calls for TPC-C txns.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.42.000 - changed w_id fields from short to long to support >32K warehouses
 *                  4.20.000 - updated rev number to match kit
 *                  4.10.001 - not deleting error class in catch handler on deadlock retry;
 *                                        not a functional bug, but a memory leak
 *                                   - had to tweak some declarations to compile with latest SDK; no functional
change
 */

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

#define DBNTWIN32
#include <sqlfront.h>
#include <sqldb.h>

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

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

#include "..\..\common\src\error.h"
```

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

#define   DEFCLPACKSIZE                         4096

// version string; must match return value from tpcc_version stored proc
const char          sVersion[] = "4.10.000";

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

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

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
          switch( ul_reason_for_call )
          {
                    case DLL_PROCESS_ATTACH:
                              DisableThreadLibraryCalls(hModule);
                              dbinit();           // initialize dblib
                              break;

                    case DLL_PROCESS_DETACH:
                              dbexit();           // close all dblib structures/connections
                              break;

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


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

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

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

/* FUNCTION: int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity, char *msgtext)
 *
 * PURPOSE:         This function handles DB-Library SQL Server error messages
 *
 * ARGUMENTS:       DBPROCESS           *dbproc                     DBPROCESS id pointer
 *                                      DBINT                       msgno                         message number
 *                                      int                                 msgstate            message state
 *                                      int                                 severity            message severity
 *                                      char                        *msgtext            printable message description
 *
 * RETURNS:                  int                                   INT_CONTINUE        continue if error is SQLETIME
else INT_CANCEL action
 *                                                                            INT_CANCEL                    cancel
operation
 *
 * COMMENTS:        This function also sets the dead lock dbproc variable if necessary.
 *
 */

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

int msg_handler(DBPROCESS *dbproc, DBINT msgno, int msgstate, int severity,
                                      LPCSTR msgtext, LPCSTR srvname, LPCSTR procname, DBUSMALLINT line)
{
          CTPCC_DBLIB                           *pConn;

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

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

          return 0;
}

/* FUNCTION: void UtilStrCpy(char * pDest, char * pSrc, int n)
 *
 * PURPOSE:         This function copies n characters from string pSrc to pDst and places a
 *                  null character at the end of the destination string.
 *
 * ARGUMENTS:       char                        *pDest    destination string pointer
 *                                      char                        *pSrc     source string pointer
```

```
 *                                      int                                     n                   number of
characters to copy
 *
 * RETURNS:                   None
 *
 * COMMENTS:        Unlike strncpy this function ensures that the result string is
 *                                      always null terminated.
 *
 */

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

          return;
}

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
 *
 */

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

          static SERRORMSG errorMsgs[] =
          {
                    { ERR_WRONG_SP_VERSION,                 "Wrong version of stored procs on database server"
          },
                    { ERR_INVALID_CUST,                     "Invalid Customer id,name."
                              },
                    { ERR_NO_SUCH_ORDER,                    "No orders found for customer."
                              },
                    { ERR_RETRIED_TRANS,                    "Retries before transaction succeeded."
          },
                    { 0,                                                                    ""
                                                                                                        }
          };

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_DBLIB* CTPCC_DBLIB_new(
          LPCSTR szServer,              // name of SQL server
          LPCSTR szUser,                          // user name for login
          LPCSTR szPassword,            // password for login
          LPCSTR szHost,                          // workstation name; shows up in sp_who; max 30 chars, only first 10
kept by SQL Server
          LPCSTR szDatabase )           // name of database to use
{
          return new CTPCC_DBLIB( szServer, szUser, szPassword, szHost, szDatabase );
}


CTPCC_DBLIB::CTPCC_DBLIB (
          LPCSTR szServer,              // name of SQL server
          LPCSTR szUser,                          // user name for login
          LPCSTR szPassword,            // password for login
          LPCSTR szHost,                          // workstation name; shows up in sp_who; max 30 chars, only first 10
kept by SQL Server
          LPCSTR szDatabase )           // name of database to use
{
          LOGINREC  *login;
          const BYTE          *pData;

          // initialization
          m_dbproc = NULL;
          m_DbLibErr = (CDBLIBERR*)NULL;
          m_SqlErr = (CSQLERR*)NULL;

          m_MaxRetries = 10;            // how many retries on deadlock

          // increase max number of connections if getting close
          if ( dbgetmaxprocs() < (iConnectionCount+5) )
          {
                    if ( dbsetmaxprocs(iConnectionCount+10) == FAIL )
                              ThrowError(CDBLIBERR::eDbSetMaxProcs);
          }

          // allocate a login structure
          login = dblogin();
```

```
	if (login == NULL)
		ThrowError(CDBLIBERR::eLogin);
	InterlockedIncrement( &iConnectionCount );

	// register error and message handler functions
	if (dbprocerrhandle(login, err_handler) == NULL)
		ThrowError(CDBLIBERR::eDbProcHandler);

	if (dbprocmsghandle(login, msg_handler) == NULL)
		ThrowError(CDBLIBERR::eDbProcHandler);


	DBSETLUSER(login, szUser);
	DBSETLPWD(login, szPassword);
	DBSETLHOST(login, szHost);
	DBSETLPACKET(login, (unsigned short)DEFCLPACKSIZE);
	DBSETLVERSION(login, DBVER60);			// use dblib ver 6.0 client behavior

	// set time to wait for login
	if (dbsetlogintime(60) == FAIL)
		ThrowError(CDBLIBERR::eDbSet);

	// set time to wait for statement execution
	if (dbsettime(180) == FAIL)
		ThrowError(CDBLIBERR::eDbSet);

	m_dbproc = dbopen(login, szServer);

	// deallocate login structure before checking for success
	dbfreelogin( login );

	if (m_dbproc == NULL)
		ThrowError(CDBLIBERR::eDbOpen);

	// save address of class instance so that the message and error handler
	// can get to data.
	dbsetuserdata(m_dbproc, (LPVOID)this);

	// Use the the right database
	if (dbuse(m_dbproc, szDatabase) == FAIL)
		ThrowError(CDBLIBERR::eDbUse);

	dbcmd(m_dbproc, "set nocount on ");			// do not return row counts
	dbcmd(m_dbproc, "set XACT_ABORT ON");		// rollback transaction on abort

	if (dbsqlexec(m_dbproc) == FAIL)
		ThrowError(CDBLIBERR::eDbSqlExec);

	DiscardNextResults(2);

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

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

	if (dbresults(m_dbproc) != SUCCEED)
		ThrowError(CDBLIBERR::eDbResults);

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

	char szSrvVersion[16];
	pData=dbdata(m_dbproc, 1);
	if (pData)
		UtilStrCpy(szSrvVersion, pData, dbdatlen(m_dbproc, 1));
	else
		szSrvVersion[0]=0;
	if (strcmp(szSrvVersion,sVersion))
		throw new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_WRONG_SP_VERSION );


	DiscardNextRows(0);
	DiscardNextResults(0);
}


CTPCC_DBLIB::~CTPCC_DBLIB( void )
{
	// close db connection and deallocate resources
	dbclose(m_dbproc);
	InterlockedDecrement( &iConnectionCount );
	if (m_DbLibErr != NULL)
		delete m_DbLibErr;
	if (m_SqlErr != NULL)
		delete m_SqlErr;
}

void CTPCC_DBLIB::SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr)
{
	delete m_DbLibErr;
	m_DbLibErr = new CDBLIBERR(CDBLIBERR::eUnknown, severity, dberr, oserr);
```

```
          if (dberrstr != NULL)
          {
                    m_DbLibErr->m_dberrstr = new char[ strlen(dberrstr)+1 ];
                    strcpy( m_DbLibErr->m_dberrstr, dberrstr );
          }

          if (oserrstr != NULL)
          {
                    m_DbLibErr->m_oserrstr = new char[ strlen(oserrstr)+1 ];
                    strcpy( m_DbLibErr->m_oserrstr, oserrstr );
          }

}

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int severity, LPCSTR msgtext )
{
          if (m_SqlErr == NULL)
                    m_SqlErr = new CSQLERR();

          m_SqlErr->m_msgno = msgno;
          m_SqlErr->m_msgstate = msgstate;
          m_SqlErr->m_severity = severity;

          delete [] m_SqlErr->m_msgtext;
          if (msgtext != NULL)
          {
                    m_SqlErr->m_msgtext = new char[ strlen(msgtext)+1 ];
                    strcpy( m_SqlErr->m_msgtext, msgtext );
          }
}

void CTPCC_DBLIB::ThrowError( CDBLIBERR::ACTION eAction )
{
          // discard anything still in return buffer
          DiscardNextRows(-1);
          DiscardNextResults(-1);

          // check for SQL Server error first;  if yes, throw it and ignore any DBLib error.
          if (m_SqlErr != NULL)
          {
                    CSQLERR             *pSqlErr;
                    pSqlErr = m_SqlErr;
                    m_SqlErr = NULL;    // clear our pointer to instance; catch handler will delete
                    throw pSqlErr;
          }

          CDBLIBERR *pDbLibErr;
          if (m_DbLibErr == NULL)
                    // this case isn't expected to happen, since it means that an error was returned
                    // but the error handlers were not called.
                    pDbLibErr = new CDBLIBERR(eAction);
          else
          {
                    pDbLibErr = m_DbLibErr;
                    pDbLibErr->m_eAction = eAction;
                    m_DbLibErr = NULL;            // clear our pointer to instance; catch handler will delete
          }

          throw pDbLibErr;
}

// Read and discard rows until no more.  Throw an exception if number of rows read doesn't
// match number of rows expected.  The row count will be ignored if the expected count value
// passed in is negative.  A typical use of this routine is to verify that there are no more
// rows to be read.
void CTPCC_DBLIB::DiscardNextRows(int iExpectedCount)
{
          int                 iRowsRead = 0;
          RETCODE   rc;

          while (TRUE)
          {
                    rc = dbnextrow(m_dbproc);
                    if (rc == NO_MORE_ROWS)
                              break;
                    if (rc == FAIL)
                    {
                              if (iExpectedCount >= 0)
                                        ThrowError(CDBLIBERR::eDbNextRow);
                              else
                                        break;
                    }
                    iRowsRead++;
          }

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

// Read and discard results until no more.  Throw an exception if number of result sets read doesn't
// match number expected.  The result set count will be ignored if the expected count value
// passed in is negative.  A typical use of this routine is to verify that there are no more
```

```
// result sets to be read.
void CTPCC_DBLIB::DiscardNextResults(int iExpectedCount)
{
          int                 iResultsRead = 0;
          RETCODE   rc;

          while (TRUE)
          {
                    rc = dbresults(m_dbproc);
                    if (rc == NO_MORE_RESULTS)
                              break;
                    if (rc == FAIL)
                    {
                              if (iExpectedCount >= 0)
                                        ThrowError(CDBLIBERR::eDbResults);
                              else
                                        break;
                    }

                    DiscardNextRows(-1);
                    iResultsRead++;
          }

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

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

          ResetError();

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

                              dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.StockLevel.w_id);
          // @w_id int
                              dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.StockLevel.d_id);
          // @d_id tinyint
                              dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *) &m_txn.StockLevel.threshold);
// @threshhold smallint

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

                              if (dbresults(m_dbproc) != SUCCEED)
                                        ThrowError(CDBLIBERR::eDbResults);

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

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

                              DiscardNextRows(0);
                              DiscardNextResults(0);

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

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


void CTPCC_DBLIB::NewOrder()
{
          int                           i;
          DBINT               commit_flag;
          DBDATETIME          datetime;
          DBDATEREC daterec;
```

```
	int			iTryCount = 0;
	const BYTE		*pData;

	ResetError();

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

			dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.w_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.d_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.NewOrder.c_id);
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_ol_cnt);

			// check whether any order lines are for a remote warehouse
			m_txn.NewOrder.o_all_local = 1;
			for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
			{
				if (m_txn.NewOrder.OL[i].ol_supply_w_id != m_txn.NewOrder.w_id)
				{
					m_txn.NewOrder.o_all_local = 0;  // at least one remote warehouse
					break;
				}
			}
			dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.NewOrder.o_all_local);

			for (i = 0; i < m_txn.NewOrder.o_ol_cnt; i++)
			{
				dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_i_id);
				dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_supply_w_id);
				dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -1, -1, (BYTE *)
&m_txn.NewOrder.OL[i].ol_quantity);
			}

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


			// Get order line results
			m_txn.NewOrder.total_amount = 0;
			for (i = 0; i<m_txn.NewOrder.o_ol_cnt; i++)
			{
				if (dbresults(m_dbproc) != SUCCEED)
					ThrowError(CDBLIBERR::eDbResults);

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

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

				if(pData=dbdata(m_dbproc, 1))
					UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name, pData, dbdatlen(m_dbproc,
1));
				if(pData=dbdata(m_dbproc, 2))
					m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *) pData);
				if(pData=dbdata(m_dbproc, 3))
					UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic, pData,
dbdatlen(m_dbproc, 3));
				if(pData=dbdata(m_dbproc, 4))
					dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,4),
					SQLFLT8, (BYTE *)&m_txn.NewOrder.OL[i].ol_i_price, 8);
				if(pData=dbdata(m_dbproc, 5))

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

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

				DiscardNextRows(0);
			}


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

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

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

			if (pData=dbdata(m_dbproc, 1))
```

```
                                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,1), SQLFLT8,
(BYTE *)&m_txn.NewOrder.w_tax, 8);
                                if (pData=dbdata(m_dbproc, 2))

                                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,2), SQLFLT8,
(BYTE *)&m_txn.NewOrder.d_tax, 8);
                                if (pData=dbdata(m_dbproc, 3))
                                        m_txn.NewOrder.o_id = (*(DBINT *) pData);
                                if (pData=dbdata(m_dbproc, 4))
                                        UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc, 4));
                                if (pData=dbdata(m_dbproc, 5))
                                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,5), SQLFLT8,
(BYTE *)&m_txn.NewOrder.c_discount, 8);
                                if (pData=dbdata(m_dbproc, 6))
                                        UtilStrCpy(m_txn.NewOrder.c_credit, pData, dbdatlen(m_dbproc, 6));
                                if (pData=dbdata(m_dbproc, 7))
                                {
                                        datetime = *((DBDATETIME *) pData);
                                        dbdatecrack(m_dbproc, &daterec, &datetime);
                                        m_txn.NewOrder.o_entry_d.year   = daterec.year;
                                        m_txn.NewOrder.o_entry_d.month  = daterec.month;
                                        m_txn.NewOrder.o_entry_d.day    = daterec.day;
                                        m_txn.NewOrder.o_entry_d.hour   = daterec.hour;
                                        m_txn.NewOrder.o_entry_d.minute = daterec.minute;
                                        m_txn.NewOrder.o_entry_d.second = daterec.second;
                                }
                                if (pData=dbdata(m_dbproc, 8))
                                        commit_flag = (*(DBTINYINT *) pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);


                                if (commit_flag == 1)
                                {
                                        m_txn.NewOrder.total_amount *= ((1 + m_txn.NewOrder.w_tax +
m_txn.NewOrder.d_tax) * (1 - m_txn.NewOrder.c_discount));
                                        m_txn.NewOrder.exec_status_code = eOK;
                                }
                                else
                                        m_txn.NewOrder.exec_status_code = eInvalidItem;

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

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


void CTPCC_DBLIB::Payment()
{
        DBDATETIME          datetime;
        DBDATEREC daterec;

        int                         iTryCount = 0;
        const BYTE          *pData;

        ResetError();

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

                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLFLT8, -1, -1, (BYTE *) &m_txn.Payment.h_amount);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Payment.c_d_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Payment.c_id);

                        // if customer id is zero, then payment is by name
                        if (m_txn.Payment.c_id == 0)
                                dbrpcparam(m_dbproc, NULL, 0, SQLCHAR, -1, strlen(m_txn.Payment.c_last),
(unsigned char *)m_txn.Payment.c_last);
```

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

                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

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

                        if (dbnumcols(m_dbproc) != 27)
                                ThrowError(CDBLIBERR::eWrongNumCols);

                        if (pData=dbdata(m_dbproc, 1))
                                m_txn.Payment.c_id = *((DBINT *) pData);
                        if (pData=dbdata(m_dbproc, 2))
                                UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
                        if (pData=dbdata(m_dbproc, 3))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.Payment.h_date.year   = daterec.year;
                                m_txn.Payment.h_date.month  = daterec.month;
                                m_txn.Payment.h_date.day    = daterec.day;
                                m_txn.Payment.h_date.hour   = daterec.hour;
                                m_txn.Payment.h_date.minute = daterec.minute;
                                m_txn.Payment.h_date.second = daterec.second;
                        }
                        if (pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc, 4));
                        if (pData=dbdata(m_dbproc, 5))
                                UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc, 5));
                        if (pData=dbdata(m_dbproc, 6))
                                UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
                        if (pData=dbdata(m_dbproc, 7))
                                UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
                        if (pData=dbdata(m_dbproc, 8))
                                UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
                        if (pData=dbdata(m_dbproc, 9))
                                UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc, 9));
                        if (pData=dbdata(m_dbproc, 10))
                                UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc, 10));
                        if (pData=dbdata(m_dbproc, 11))
                                UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
                        if (pData=dbdata(m_dbproc, 12))
                                UtilStrCpy(m_txn.Payment.d_state, pData, dbdatlen(m_dbproc, 12));
                        if (pData=dbdata(m_dbproc, 13))
                                UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
                        if (pData=dbdata(m_dbproc, 14))
                                UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
                        if (pData=dbdata(m_dbproc, 15))
                                UtilStrCpy(m_txn.Payment.c_middle, pData, dbdatlen(m_dbproc, 15));
                        if (pData=dbdata(m_dbproc, 16))
                                UtilStrCpy(m_txn.Payment.c_street_1, pData, dbdatlen(m_dbproc, 16));
                        if (pData=dbdata(m_dbproc, 17))
                                UtilStrCpy(m_txn.Payment.c_street_2, pData, dbdatlen(m_dbproc, 17));
                        if (pData=dbdata(m_dbproc, 18))
                                UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
                        if (pData=dbdata(m_dbproc, 19))
                                UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc, 19));
                        if (pData=dbdata(m_dbproc, 20))
                                UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
                        if (pData=dbdata(m_dbproc, 21))
                                UtilStrCpy(m_txn.Payment.c_phone, pData, dbdatlen(m_dbproc, 21));
                        if (pData=dbdata(m_dbproc, 22))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.Payment.c_since.year   = daterec.year;
                                m_txn.Payment.c_since.month  = daterec.month;
                                m_txn.Payment.c_since.day    = daterec.day;
                                m_txn.Payment.c_since.hour   = daterec.hour;
                                m_txn.Payment.c_since.minute = daterec.minute;
                                m_txn.Payment.c_since.second = daterec.second;
                        }
                        if(pData=dbdata(m_dbproc, 23))
                                UtilStrCpy(m_txn.Payment.c_credit, pData, dbdatlen(m_dbproc, 23));
                        if(pData=dbdata(m_dbproc, 24))
                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,24), SQLFLT8,
(BYTE *)&m_txn.Payment.c_credit_lim, 8);
                        if(pData=dbdata(m_dbproc, 25))
                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,25), SQLFLT8,
(BYTE *)&m_txn.Payment.c_discount, 8);
                        if(pData=dbdata(m_dbproc, 26))
                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,26), SQLFLT8,
(BYTE *)&m_txn.Payment.c_balance, 8);
                        if(pData=dbdata(m_dbproc, 27))
                                UtilStrCpy(m_txn.Payment.c_data, pData, dbdatlen(m_dbproc, 27));

                        DiscardNextRows(0);
                        DiscardNextResults(0);


                        if (m_txn.Payment.c_id == 0)
```

```
                                        throw new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_INVALID_CUST );
                                else
                                        m_txn.Payment.exec_status_code = eOK;

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

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



void CTPCC_DBLIB::OrderStatus()
{
        int                             i;
        DBDATETIME          datetime;
        DBDATEREC daterec;

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

        ResetError();

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

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

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


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

                        // Get order lines
                        if (dbresults(m_dbproc) != SUCCEED)
                        {
                                if ((m_DbLibErr == NULL) && (m_SqlErr == NULL))
                                        throw new CTPCC_DBLIB_ERR( CTPCC_DBLIB_ERR::ERR_NO_SUCH_ORDER );
                                else
                                        ThrowError(CDBLIBERR::eDbResults);
                        }

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

                        i = 0;
                        while (TRUE)
                        {
                                rc = dbnextrow(m_dbproc);
                                if (rc == NO_MORE_ROWS)
                                        break;
                                if (rc != REG_ROW)
                                        ThrowError(CDBLIBERR::eDbNextRow);

                                if(pData=dbdata(m_dbproc, 1))
                                        m_txn.OrderStatus.OL[i].ol_supply_w_id = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 2))
                                        m_txn.OrderStatus.OL[i].ol_i_id = (*(DBINT *) pData);
                                if(pData=dbdata(m_dbproc, 3))
                                        m_txn.OrderStatus.OL[i].ol_quantity = (*(DBSMALLINT *) pData);
                                if(pData=dbdata(m_dbproc, 4))
                                        dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,4),
                                                SQLFLT8, (BYTE
*)&m_txn.OrderStatus.OL[i].ol_amount, 8);
                                if(pData=dbdata(m_dbproc, 5))
                                {
                                        datetime = *((DBDATETIME *) pData);
```

```
                                        dbdatecrack(m_dbproc, &daterec, &datetime);
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.year   = daterec.year;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.month  = daterec.month;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.day    = daterec.day;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.hour   = daterec.hour;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.minute = daterec.minute;
                                        m_txn.OrderStatus.OL[i].ol_delivery_d.second = daterec.second;
                                }
                                i++;
                        }
                        m_txn.OrderStatus.o_ol_cnt = i;


                        if (dbresults(m_dbproc) != SUCCEED)
                                ThrowError(CDBLIBERR::eDbResults);

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

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

                        if(pData=dbdata(m_dbproc, 1))
                                m_txn.OrderStatus.c_id = (*(DBINT *) pData);
                        if(pData=dbdata(m_dbproc, 2))
                                UtilStrCpy(m_txn.OrderStatus.c_last, pData, dbdatlen(m_dbproc,2));
                        if(pData=dbdata(m_dbproc, 3))
                                UtilStrCpy(m_txn.OrderStatus.c_first, pData, dbdatlen(m_dbproc,3));
                        if(pData=dbdata(m_dbproc, 4))
                                UtilStrCpy(m_txn.OrderStatus.c_middle, pData, dbdatlen(m_dbproc, 4));
                        if(pData=dbdata(m_dbproc, 5))
                        {
                                datetime = *((DBDATETIME *) pData);
                                dbdatecrack(m_dbproc, &daterec, &datetime);
                                m_txn.OrderStatus.o_entry_d.year   = daterec.year;
                                m_txn.OrderStatus.o_entry_d.month  = daterec.month;
                                m_txn.OrderStatus.o_entry_d.day    = daterec.day;
                                m_txn.OrderStatus.o_entry_d.hour   = daterec.hour;
                                m_txn.OrderStatus.o_entry_d.minute = daterec.minute;
                                m_txn.OrderStatus.o_entry_d.second = daterec.second;
                        }
                        if(pData=dbdata(m_dbproc, 6))
                                m_txn.OrderStatus.o_carrier_id = (*(DBSMALLINT *) pData);
                        if(pData=dbdata(m_dbproc, 7))
                                dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData, dbdatlen(m_dbproc,7),
                                                SQLFLT8, (BYTE *)&m_txn.OrderStatus.c_balance, 8);
                        if(pData=dbdata(m_dbproc, 8))
                                m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                        DiscardNextRows(0);
                        DiscardNextResults(0);


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

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

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


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

        ResetError();

        while (TRUE)
        {
```

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

                              dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *) &m_txn.Delivery.w_id);
                              dbrpcparam(m_dbproc, NULL, 0, SQLINT1, -1, -1, (BYTE *) &m_txn.Delivery.o_carrier_id);

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

                              if (dbresults(m_dbproc) != SUCCEED)
                                        ThrowError(CDBLIBERR::eDbResults);

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

                              if (dbnumcols(m_dbproc) != 10)
                                        ThrowError(CDBLIBERR::eWrongNumCols);

                              for (i=0; i<10; i++)
                              {
                                        if (pData = dbdata(m_dbproc, i+1))
                                                  m_txn.Delivery.o_id[i] = *((DBINT *)pData);
                              }

                              DiscardNextRows(0);
                              DiscardNextResults(0);

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

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

void CTPCC_DBLIB::ResetError()
{
          if (m_DbLibErr != NULL)
          {
                    delete m_DbLibErr;
                    m_DbLibErr = (CDBLIBERR*)NULL;
          }

          if (m_SqlErr != NULL)
          {
                    delete m_SqlErr;
                    m_SqlErr = (CSQLERR*)NULL;
          }
          return;
}
```

db_dblib_dll/src/tpcc_dblib.h

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

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

// need to declare functions for import, unless define has already been created
```

```
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

class CSQLERR : public CBaseErr
{
          public:

                    CSQLERR(void)
                    {
                              m_msgno = 0;
                              m_msgstate = 0;
                              m_severity = 0;
                              m_msgtext = NULL;
                    };

                    ~CSQLERR()
                    {
                              delete [] m_msgtext;
                    };

                    int                m_msgno;
                    int                m_msgstate;
                    int                m_severity;
                    char   *m_msgtext;

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

};

class CDBLIBERR : public CBaseErr
{
          public:
                    enum ACTION
                    {
                              eNone,
                              eUnknown,
                              eLogin,                                   // error from dblogin
                              eDbOpen,                        // error from dbopen
                              eDbUse,                                   // error from dbuse
                              eDbSqlExec,                               // error from dbsqlexec
                              eDbSet,                                   // error from one of the dbset* routines
                              eDbNextRow,                               // error from dbnextrow
                              eWrongRowCount,                 // more or less rows returned than expected
                              eWrongNumCols,                  // more or less columns returned than expected
                              eDbResults,                               // error from dbresults
                              eDbRpcExec,                               // error from dbrpcexec
                              eDbSetMaxProcs,                 // error from dbsetmaxprocs
                              eDbProcHandler                  // error from either dbprocerrhandle or dbprocmsghandle
                    };

                    CDBLIBERR(ACTION eAction, int severity = 0, int dberror = 0, int oserr = 0)
                    {
                              m_eAction = eAction;
                              m_severity = severity;
                              m_dberror = dberror;
                              m_oserr = oserr;

                              m_dberrstr = NULL;
                              m_oserrstr = NULL;
                    };

                    ~CDBLIBERR()
                    {
                              delete [] m_dberrstr;
                              delete [] m_oserrstr;
                    };

                    ACTION    m_eAction;
                    int                m_severity;
                    int                m_dberror;
                    int                m_oserr;
                    char   *m_dberrstr;
                    char   *m_oserrstr;

                    int ErrorType() {return ERR_TYPE_DBLIB;};
                    int ErrorNum() {return m_dberror;};
                    char *ErrorText() {return m_dberrstr;};

};

class CTPCC_DBLIB_ERR : public CBaseErr
{
          public:
                    enum CTPCC_DBLIB_ERRS
                    {
                              ERR_WRONG_SP_VERSION = 1,     // "Wrong version of stored procs on database server"
                              ERR_INVALID_CUST,                         // "Invalid Customer id,name."
                              ERR_NO_SUCH_ORDER                         // "No orders found for customer."
                    };
```

```
                    CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; };

                    int                 m_errno;

                    int ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
                    int ErrorNum() {return m_errno;};

                    char *ErrorText();
};

class DllDecl CTPCC_DBLIB : public CTPCC_BASE
{
          private:
                    // declare variables and private functions here...
                    PDBPROCESS          m_dbproc;
                    CDBLIBERR *m_DbLibErr;                  // not allocated until needed (maybe never)
                    CSQLERR             *m_SqlErr;                              // not allocated until needed (maybe
never)
                    int                           m_MaxRetries;                 // retry count on deadlock

                    void DiscardNextRows(int iExpectedCount);
                    void DiscardNextResults(int iExpectedCount);
                    void ThrowError( CDBLIBERR::ACTION eAction );
                    void ResetError();

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

          public:
                    CTPCC_DBLIB(LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );
                    ~CTPCC_DBLIB(void);

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

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


                    // these are public because they must be called from the dblib err_handler and msg_hangler
                    // outside of the class
                    void SetDbLibError(int severity, int dberr, int oserr, LPCSTR dberrstr, LPCSTR oserrstr);
                    void SetSqlError( int msgno, int msgstate, int severity, LPCSTR msgtext );

};

extern "C" DllDecl CTPCC_DBLIB* CTPCC_DBLIB_new
          ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR szDatabase );

typedef CTPCC_DBLIB* (TYPE_CTPCC_DBLIB)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

tm_com_dll/src/tpcc_com.cpp

```
/*        FILE:               TPCC_COM.CPP
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                    not yet audited
 *
 *        PURPOSE:  Source file for TPC-C COM+ class implementation.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.20.000 - first version
 */

// needed for CoinitializeEx
#define _WIN32_WINNT 0x0400

#include <windows.h>

// need to declare functions for export
```

```
#define DllDecl  __declspec( dllexport )

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

#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\tpcc_com_all\src\tpcc_com_all_i.c"

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL bSinglePool)
{
          return new CTPCC_COM(bSinglePool);
}

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

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

          m_bSinglePool       = bSinglePool;

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

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

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

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

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

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

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

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

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

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

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

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

                    hr = m_pStockLevel->CallSetComplete();
```

```
                    if (FAILED(hr))
                              throw new CCOMERR(hr);

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

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

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

void CTPCC_COM::NewOrder()
{
          VARIANT   vTxn_out;

          HRESULT hr = m_pNewOrder->NewOrder(m_vTxn, &vTxn_out);
          if (FAILED(hr))
                    throw new CCOMERR( hr );
          memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray->rgsabound[0].cElements);
          SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::Payment()
{
          VARIANT   vTxn_out;

          HRESULT hr = m_pPayment->Payment(m_vTxn, &vTxn_out);
          if (FAILED(hr))
                    throw new CCOMERR( hr );
          memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray->rgsabound[0].cElements);
          SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::StockLevel()
{
          VARIANT   vTxn_out;

          HRESULT hr = m_pStockLevel->StockLevel(m_vTxn, &vTxn_out);
          if (FAILED(hr))
                    throw new CCOMERR( hr );
          memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray->rgsabound[0].cElements);
          SafeArrayDestroy(vTxn_out.parray);

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

void CTPCC_COM::OrderStatus()
{
          VARIANT   vTxn_out;

          HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn, &vTxn_out);
          if (FAILED(hr))
                    throw new CCOMERR( hr );
          memcpy(m_pTxn, (void *)vTxn_out.parray->pvData,vTxn_out.parray->rgsabound[0].cElements);
          SafeArrayDestroy(vTxn_out.parray);

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

tm_com_dll/src/tpcc_com.h

```
/*        FILE:               TPCC_COM.H
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                    not yet audited
 *
```

```
 *        PURPOSE:  Header file for TPC-C COM+ class implementation.
 *
 *  Change history:
 *                  4.20.000 - first version
 */

#pragma once

#include <stdio.h>
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifndef DllDecl
#define DllDecl  __declspec( dllimport )
#endif

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

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

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

                    int                m_hr;
                    int                m_iErrorType;
                    int                m_iError;

                    // A CCOMERR class can impersonate another class, which happens if the error
                    // was not actually a COM Services error, but was simply transmitted back via COM.
                    int ErrorType()
                    {
                              if (m_iErrorType == 0)
                                        return ERR_TYPE_COM;
                              else
                                        return m_iErrorType;
                    }

                    int ErrorNum() {return m_hr;}

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

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

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

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

                    VARIANT m_vTxn;
          public:
```

```
                    CTPCC_COM(BOOL bSinglePool);
                    ~CTPCC_COM(void);

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

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


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


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

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);
```

tpcc_com_all/src/methods.h

```
/*        FILE:               METHODS.H
 *                                  Microsoft TPC-C Kit Ver. 4.20.000
 *                                  Copyright Microsoft, 1999
 *              All Rights Reserved
 *
 *                                  not yet audited
 *
 *        PURPOSE:  Header file for COM components.
 *
 *  Change history:
 *                  4.20.000 - first version
 */


enum COMPONENT_ERROR
{
          ERR_MISSING_REGISTRY_ENTRIES = 1,
          ERR_LOADDLL_FAILED,
          ERR_GETPROCADDR_FAILED,
          ERR_UNKNOWN_DB_PROTOCOL
};

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

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

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

                    COMPONENT_ERROR     m_Error;
                    char                          *m_szTextDetail;
```

```
                    char                        *m_szErrorText;
                    DWORD                       m_SystemErr;

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


static void WriteMessageToEventLog(LPTSTR lpszMsg);


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

          CTPCC_Common();
          ~CTPCC_Common();

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

          HRESULT __stdcall CallSetComplete();

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

// IObjectConstruct
          STDMETHODIMP Construct(IDispatch * pUnk);

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

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

};

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

BEGIN_COM_MAP(CTPCC)
          COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
          COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};


/////////////////////////////////////////////////////////////////////////////
// CNewOrder
class CNewOrder :
          public CTPCC_Common,
          public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
```

```
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

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

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


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

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

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


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

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

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


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

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

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

tpcc_com_all/src/resource.h

```
//{{NO_DEPENDENCIES}}
```

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

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE         32768
#define _APS_NEXT_CONTROL_VALUE         201
#define _APS_NEXT_SYMED_VALUE           106
#endif
#endif
```

tpcc_com_all/src/tpcc_com_all.cpp

```
/*        FILE:               TPCC_COM_ALL.CPP
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *              All Rights Reserved
 *
 *                                    Version 4.10.000 audited by Richard Gimarc, Performance Metrics, 3/17/99
 *
 *        PURPOSE:  Implementation for TPC-C Tuxedo class.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  4.20.000 - updated rev number to match kit
 */

#define STRICT
#define _WIN32_WINNT 0x0400
#define _ATL_APARTMENT_THREADED

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

#include <atlcom.h>
#include <initguid.h>
#include <transact.h>
#include <atlimpl.cpp>
#include <comsvcs.h>

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

#include "tpcc_com_ps.h"
#include "..\..\common\src\trans.h"                                          //tpckit transaction header contains
definations of structures specific to TPC-C
#include "..\..\common\src\txn_base.h"
#include "..\..\common\src\error.h"
#include "..\..\common\src\ReadRegistry.h"
#include "..\..\db_dblib_dll\src\tpcc_dblib.h"              // DBLIB implementation of TPC-C txns
#include "..\..\db_odbc_dll\src\tpcc_odbc.h"                // ODBC implementation of TPC-C txns

#include "resource.h"
#include "tpcc_com_all.h"
#include "tpcc_com_all_i.c"
#include "Methods.h"
#include "..\..\tpcc_com_ps\src\tpcc_com_ps_i.c"
#include "..\..\common\src\ReadRegistry.cpp"


CComModule _Module;

BEGIN_OBJECT_MAP(ObjectMap)
          OBJECT_ENTRY(CLSID_TPCC, CTPCC)
          OBJECT_ENTRY(CLSID_NewOrder, CNewOrder)
          OBJECT_ENTRY(CLSID_OrderStatus, COrderStatus)
          OBJECT_ENTRY(CLSID_Payment, CPayment)
          OBJECT_ENTRY(CLSID_StockLevel, CStockLevel)
END_OBJECT_MAP()


// configuration settings from registry
TPCCREGISTRYDATA    Reg;
char                                szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];
```

```
static  HINSTANCE hLibInstanceDb = NULL;

TYPE_CTPCC_DBLIB    *pCTPCC_DBLIB_new;
TYPE_CTPCC_ODBC             *pCTPCC_ODBC_new;


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

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

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

                              DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
                              GetComputerName(szMyComputerName, &dwSize);
                              szMyComputerName[dwSize] = 0;

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

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

                                        // get function pointer to wrapper for class constructor
                                        pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
                                        if (pCTPCC_DBLIB_new == NULL)
                                                  throw new CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                              }
                              else if (Reg.eDB_Protocol == ODBC)
                              {
                                        strcpy( szDllName, Reg.szPath );
                                        strcat( szDllName, "tpcc_odbc.dll");
                                        hLibInstanceDb = LoadLibrary( szDllName );
                                        if (hLibInstanceDb == NULL)
                                                  throw new CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName,
GetLastError() );

                                        // get function pointer to wrapper for class constructor
                                        pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
                                        if (pCTPCC_ODBC_new == NULL)
                                                  throw new CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName,
GetLastError() );
                              }
                              else
                                        throw new CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );
                    }
                    else if (dwReason == DLL_PROCESS_DETACH)
                              _Module.Term();

          }
          catch (CBaseErr *e)
          {
                    WriteMessageToEventLog(e->ErrorText());
                    delete e;
                    return FALSE;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception in object DllMain"));
                    return FALSE;
          }

          return TRUE;        // OK
}

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

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

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

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

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

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

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

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


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

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

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

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

        (VOID) DeregisterEventSource(hEventSource);
    }
}

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

/* FUNCTION: CCOMPONENT_ERR::ErrorText
 *
 */

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

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

```
                    {
                              strcpy( szTmp, errorMsgs[i].szMsg );
                              break;
                    }
                    i++;
          }

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

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


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

CTPCC_Common::~CTPCC_Common()
{
          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
          {
                    if (Reg.eDB_Protocol == ODBC)
                              m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
                    else if (Reg.eDB_Protocol == DBLIB)
                              m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
szMyComputerName, Reg.szDbName );
          }
          catch (CBaseErr *e)
          {
                    WriteMessageToEventLog(e->ErrorText());
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception in object ::Construct"));
                    return E_FAIL;
          }

          return S_OK;
}

HRESULT CTPCC_Common::NewOrder(VARIANT txn_in, VARIANT* txn_out)
{
          PNEW_ORDER_DATA     pNewOrder;
          COM_DATA            *pData;
          try
          {
                    pData = (COM_DATA*)txn_in.parray->pvData;
                    pNewOrder = m_pTxn->BuffAddr_NewOrder();

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

                    m_pTxn->NewOrder();           // do the actual txn

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray = SafeArrayCreateVector(VT_UI1,
```

```
                                                                txn_in.parray->rgsabound->cElements,
                                                                txn_in.parray->rgsabound->cElements);
                    pData = (COM_DATA*) txn_out->parray->pvData;

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

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database connection; if yes, component is toast
                    if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                              m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

HRESULT CTPCC_Common::Payment(VARIANT txn_in, VARIANT* txn_out)
{
          PPAYMENT_DATA       pPayment;
          COM_DATA            *pData;
          try
          {
                    pData = (COM_DATA*)txn_in.parray->pvData;
                    pPayment = m_pTxn->BuffAddr_Payment();

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

                    m_pTxn->Payment();            // do the actual txn

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray = SafeArrayCreateVector( VT_UI1,
                                                                txn_in.parray->rgsabound->cElements,
                                                                txn_in.parray->rgsabound->cElements);
                    pData = (COM_DATA*) txn_out->parray->pvData;

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

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database connection; if yes, component is toast
                    if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                              m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

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

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

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

                    m_pTxn->StockLevel();
```

```
                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray = SafeArrayCreateVector( VT_UI1,
                                                             txn_in.parray->rgsabound->cElements,
                                                             txn_in.parray->rgsabound->cElements);
                    pData = (COM_DATA*)txn_out->parray->pvData;

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

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database connection; if yes, component is toast
                    if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                             m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}

HRESULT CTPCC_Common::OrderStatus(VARIANT txn_in, VARIANT* txn_out)
{
          PORDER_STATUS_DATA  pOrderStatus;
          COM_DATA                    *pData;
          try
          {
                    pData = (COM_DATA*)txn_in.parray->pvData;
                    pOrderStatus = m_pTxn->BuffAddr_OrderStatus();

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

                    m_pTxn->OrderStatus();

                    VariantInit(txn_out);
                    txn_out->vt = VT_SAFEARRAY;
                    txn_out->parray = SafeArrayCreateVector( VT_UI1,
                                                             txn_in.parray->rgsabound->cElements,
                                                             txn_in.parray->rgsabound->cElements);
                    pData = (COM_DATA*)txn_out->parray->pvData;

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

                    pData->retval = ERR_SUCCESS;
                    pData->error = 0;
                    return S_OK;
          }
          catch (CBaseErr *e)
          {
                    // check for lost database connection; if yes, component is toast
                    if ( ((e->ErrorType() == ERR_TYPE_DBLIB) && (e->ErrorNum() == 10005)) ||
                              ((e->ErrorType() == ERR_TYPE_ODBC) && (e->ErrorNum() == 10054)) )
                             m_bCanBePooled = FALSE;

                    pData->retval = e->ErrorType();
                    pData->error = e->ErrorNum();
                    delete e;
                    return E_FAIL;
          }
          catch (...)
          {
                    WriteMessageToEventLog(TEXT("Unhandled exception."));
                    pData->retval = ERR_TYPE_LOGIC;
                    pData->error = 0;
                    m_bCanBePooled = FALSE;
                    return E_FAIL;
          }
}
```

tpcc_com_all/src/tpcc_com_all.def

```
; tpcc_com_all.def : Declares the module parameters.
```

```
LIBRARY      "tpcc_com_all.dll"

EXPORTS
          DllCanUnloadNow     @1 PRIVATE
          DllGetClassObject   @2 PRIVATE
          DllRegisterServer   @3 PRIVATE
          DllUnregisterServer @4 PRIVATE
```

tpcc_com_all/src/tpcc_com_all.h

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:18 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

#ifndef __tpcc_com_all_h__
#define __tpcc_com_all_h__

/* 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_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

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




extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;


#ifndef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

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


EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

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

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#endif /* __TPCCLib_LIBRARY_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

/* end of Additional Prototypes */
```

```
#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/tpcc_com_all.idl

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

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


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

[
          uuid(122A3117-2520-11D3-BA71-00C04FBFE08B),
          version(1.0),
          helpstring("TPC-C 1.0 Type Library")
]
library TPCCLib
{
          importlib("stdole32.tlb");
          importlib("stdole2.tlb");

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


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


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


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


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

```
        ]
        coclass StockLevel
        {
                [default] interface ITPCC;
        };

};
```

tpcc_com_all/src/tpcc_com_all.rc

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

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

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

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

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

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

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

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

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

#endif    // APSTUDIO_INVOKED


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

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

```
            VALUE "ProductVersion", "1, 0, 0, 1\0"
            VALUE "OLESelfRegister", "\0"
        END
    END
    BLOCK "VarFileInfo"
    BEGIN
        VALUE "Translation", 0x409, 1200
    END
END

#endif    // !_MAC


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

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

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

STRINGTABLE DISCARDABLE
BEGIN
    IDS_PROJNAME            "tpcc_com_all"
END

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



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

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

tpcc_com_all/src/tpcc_com_all.rgs

```
HKCR
{
          TPCC.AllTxns.1 = s 'All Txns Class'
          {
                    CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
          }
          TPCC.AllTxns = s 'TPCC Class'
          {
                    CurVer = s 'TPCC.AllTxns.1'
          }
          NoRemove CLSID
          {
                    ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC Class'
                    {
                              ProgID = s 'TPCC.AllTxns.1'
                              VersionIndependentProgID = s 'TPCC.AllTxns'
                              InprocServer32 = s '%MODULE%'
                              {
                                        val ThreadingModel = s 'Both'
                              }
                    }
          }
}
```

tpcc_com_all/src/tpcc_com_all_i.c

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

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */
```

```
/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:18 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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


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


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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


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

```
/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:18 2000
 */
/* Compiler settings for .\src\tpcc_com_all.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

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


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


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

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_all/src/tpcc_com_no.rgs

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

tpcc_com_all/src/tpcc_com_os.rgs

```
HKCR
{
	TPCC.OrderStatus.1 = s 'OrderStatus Class'
	{
		CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
	}
	TPCC.OrderStatus = s 'OrderStatus Class'
	{
		CurVer = s 'TPCC.OrderStatus.1'
	}
	NoRemove CLSID
	{
		ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s 'OrderStatus Class'
		{
			ProgID = s 'TPCC.OrderStatus.1'
			VersionIndependentProgID = s 'TPCC.OrderStatus'
			InprocServer32 = s '%MODULE%'
			{
				val ThreadingModel = s 'Both'
			}
		}
	}
}
```

tpcc_com_all/src/tpcc_com_pay.rgs

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

tpcc_com_all/src/tpcc_com_ps.h

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* 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_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

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

        virtual HRESULT __stdcall Payment(
```

```
            /* [in] */ VARIANT txn_in,
            /* [out] */ VARIANT __RPC_FAR *txn_out) = 0;

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

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

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

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else     /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

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

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

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

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

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

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

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

        END_INTERFACE
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
    };


#ifdef COBJMACROS


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

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

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


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

#define ITPCC_Payment(This,txn_in,txn_out)        \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)
```

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

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

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

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

#endif /* COBJMACROS */


#endif    /* C style interface */



HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * 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 __RPC_FAR *, unsigned long            ,
VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserMarshal(  unsigned long __RPC_FAR *, unsigned char __RPC_FAR *,
VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserUnmarshal(unsigned long __RPC_FAR *, unsigned char __RPC_FAR *,
VARIANT __RPC_FAR * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long __RPC_FAR *, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_all/src/tpcc_com_sl.rgs

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

tpcc_com_ps/src/dlldata.c

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

        DO NOT ALTER THIS FILE

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

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

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


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

```
DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

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

tpcc_com_ps/src/tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
          DllGetClassObject       @1    PRIVATE
          DllCanUnloadNow         @2    PRIVATE
          GetProxyDllInfo         @3    PRIVATE
          DllRegisterServer             @4        PRIVATE
          DllUnregisterServer           @5        PRIVATE
```

tpcc_com_ps/src/tpcc_com_ps.h

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

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


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
         __declspec(uuid()), __declspec(selectany), __declspec(novtable)
         DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING(  )


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

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

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

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

#ifndef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

/* 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_FAR * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void __RPC_FAR * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
```

```
/* [local] */



extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifndef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

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


EXTERN_C const IID IID_ITPCC;

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

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

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

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

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

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

        virtual HRESULT __stdcall CallSetComplete( void) = 0;

    };

#else     /* C style interface */

    typedef struct ITPCCVtbl
    {
        BEGIN_INTERFACE

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

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

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

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

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

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

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

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

        HRESULT ( __stdcall __RPC_FAR *CallSetComplete )(
            ITPCC __RPC_FAR * This);

        END_INTERFACE
```

```
    } ITPCCVtbl;

    interface ITPCC
    {
        CONST_VTBL struct ITPCCVtbl __RPC_FAR *lpVtbl;
    };


#ifdef COBJMACROS


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

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

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


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

#define ITPCC_Payment(This,txn_in,txn_out)        \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)

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

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

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

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

#endif /* COBJMACROS */


#endif    /* C style interface */



HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_StockLevel_Stub(
```

```
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC __RPC_FAR * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT __RPC_FAR *txn_out);


void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *_pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);


HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC __RPC_FAR * 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 __RPC_FAR *, unsigned long            ,
VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserMarshal(  unsigned long __RPC_FAR *, unsigned char __RPC_FAR *,
VARIANT __RPC_FAR * );
unsigned char __RPC_FAR * __RPC_USER  VARIANT_UserUnmarshal(unsigned long __RPC_FAR *, unsigned char __RPC_FAR *,
VARIANT __RPC_FAR * );
void                      __RPC_USER  VARIANT_UserFree(     unsigned long __RPC_FAR *, VARIANT __RPC_FAR * );

/* end of Additional Prototypes */

#ifdef __cplusplus
}
#endif

#endif
```

tpcc_com_ps/src/tpcc_com_ps.idl

```
/*        FILE:               ITPCC.IDL
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                    not yet audited
 *
 *        PURPOSE:  Defines the interface used by TPCC. This interface can be implemented by C++ components.
 *
 *  Change history:
 *                  4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

         [
                  object,
                  oleautomation,
                  uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                  helpstring("ITPCC Interface"),
                  pointer_default(unique)
         ]
         interface ITPCC : IUnknown
         {

        HRESULT _stdcall NewOrder
                                                          (
                                                          [in] VARIANT  txn_in,
```

```
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall Payment
                                                (
                                                [in] VARIANT txn_in,
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall Delivery
                                                (
                                                [in] VARIANT txn_in,
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall StockLevel
                                                (
                                                [in] VARIANT txn_in,
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall OrderStatus
                                                (
                                                [in] VARIANT txn_in,
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall CallSetComplete
                                                (
                                                );

    };    // interface ITPCC
```

tpcc_com_ps/src/tpcc_com_ps_i.c

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

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

typedef struct _IID
{
```

```
    unsigned long x;
    unsigned short s1;
    unsigned short s2;
    unsigned char  c[8];
} IID;

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif



#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


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

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_AXP64)

#ifdef __cplusplus
extern "C"{
#endif


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

#ifdef _MIDL_USE_GUIDDEF_

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

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

#else // !_MIDL_USE_GUIDDEF_

#ifndef __IID_DEFINED__
#define __IID_DEFINED__

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

#endif // __IID_DEFINED__

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

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

#endif !_MIDL_USE_GUIDDEF_

MIDL_DEFINE_GUID(IID, IID_ITPCC,0xFEEE6AA2,0x84B1,0x11d2,0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B);

#undef MIDL_DEFINE_GUID

#ifdef __cplusplus
}
#endif


#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

tpcc_com_ps/src/tpcc_com_ps_p.c

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

/* this ALWAYS GENERATED file contains the proxy stub code */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf (OptLev=i2), W1, Zp8, env=Win32 (32b run), 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_AXP64)
#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   997
#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;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* 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,0x47,0x00,0x00,0x4F,0xBF,0xE0,0x8B}} */
```

```
extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

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

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

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

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.4.280 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
```

```
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


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

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these features:
#error   -Oif or -Oicf, [wire_marshal] or [user_marshal] attribute.
#error However, your C/C++ compilation flags indicate you intend to run this app on earlier systems.
#error This app will die there with the RPC_X_WRONG_STUB_VERSION error.
#endif


static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

          /* Procedure NewOrder */

                               0x33,               /* FC_AUTO_HANDLE */
                               0x6c,               /* Old Flags:  object, Oi2 */
/*  2 */  NdrFcLong( 0x0 ),   /* 0 */
/*  6 */  NdrFcShort( 0x3 ),  /* 3 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/*  8 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                               NdrFcShort( 0x20 ), /*  MIPS Stack size/offset = 32 */
#endif
#else
                               NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                               NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 18 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset = 4 */
#else
                               NdrFcShort( 0x8 ),  /*  MIPS Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* PPC Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 20 */  NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Parameter txn_out */

/* 22 */  NdrFcShort( 0x4113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 24 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                               NdrFcShort( 0x18 ), /*  MIPS Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 26 */  NdrFcShort( 0x3da ),          /* Type Offset=986 */

          /* Return value */

/* 28 */  NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
```

```
#ifndef _PPC_
#if !defined(_MIPS_)
/* 30 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                              NdrFcShort( 0x1c ), /*  MIPS Stack size/offset = 28 */
#endif
#else
                              NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                              NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 42 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                              NdrFcShort( 0x20 ), /*  MIPS Stack size/offset = 32 */
#endif
#else
                              NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                              NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 52 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset = 4 */
#else
                              NdrFcShort( 0x8 ),  /*  MIPS Stack size/offset = 8 */
#endif
#else
                              NdrFcShort( 0x8 ),  /* PPC Stack size/offset = 8 */
#endif
#else
                              NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 54 */  NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Parameter txn_out */

/* 56 */  NdrFcShort( 0x4113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 58 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                              NdrFcShort( 0x18 ), /*  MIPS Stack size/offset = 24 */
#endif
#else
                              NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                              NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 60 */  NdrFcShort( 0x3da ),          /* Type Offset=986 */

          /* Return value */

/* 62 */  NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 64 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                              NdrFcShort( 0x1c ), /*  MIPS Stack size/offset = 28 */
#endif
#else
                              NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                              NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 76 */  NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                               NdrFcShort( 0x20 ), /*  MIPS Stack size/offset = 32 */
#endif
#else
                               NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                               NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 86 */  NdrFcShort( 0x4 ),  /* x86 Stack size/offset = 4 */
#else
                               NdrFcShort( 0x8 ),  /*  MIPS Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* PPC Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 88 */  NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

         /* Parameter txn_out */

/* 90 */  NdrFcShort( 0x4113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 92 */  NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                               NdrFcShort( 0x18 ), /*  MIPS Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 94 */  NdrFcShort( 0x3da ),          /* Type Offset=986 */

         /* Return value */

/* 96 */  NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 98 */  NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                               NdrFcShort( 0x1c ), /*  MIPS Stack size/offset = 28 */
#endif
#else
                               NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                               NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                               NdrFcShort( 0x20 ), /*  MIPS Stack size/offset = 32 */
#endif
```

```
#else
                            NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                            NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 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, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 120 */ NdrFcShort( 0x4 ),  /* x86 Stack size/offset = 4 */
#else
                            NdrFcShort( 0x8 ),  /*  MIPS Stack size/offset = 8 */
#endif
#else
                            NdrFcShort( 0x8 ),  /* PPC Stack size/offset = 8 */
#endif
#else
                            NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 122 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

         /* Parameter txn_out */

/* 124 */ NdrFcShort( 0x4113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                            NdrFcShort( 0x18 ), /*  MIPS Stack size/offset = 24 */
#endif
#else
                            NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                            NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 128 */ NdrFcShort( 0x3da ),          /* Type Offset=986 */

         /* Return value */

/* 130 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                            NdrFcShort( 0x1c ), /*  MIPS Stack size/offset = 28 */
#endif
#else
                            NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                            NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
#else
                            NdrFcShort( 0x20 ), /*  MIPS Stack size/offset = 32 */
#endif
#else
                            NdrFcShort( 0x20 ), /* PPC Stack size/offset = 32 */
#endif
#else
                            NdrFcShort( 0x28 ), /* Alpha Stack size/offset = 40 */
#endif
/* 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, */
```

```
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 154 */ NdrFcShort( 0x4 ),  /* x86 Stack size/offset = 4 */
#else
                               NdrFcShort( 0x8 ),  /*  MIPS Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* PPC Stack size/offset = 8 */
#endif
#else
                               NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 156 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Parameter txn_out */

/* 158 */ NdrFcShort( 0x4113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=16 */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 160 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
#else
                               NdrFcShort( 0x18 ), /*  MIPS Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* PPC Stack size/offset = 24 */
#endif
#else
                               NdrFcShort( 0x18 ), /* Alpha Stack size/offset = 24 */
#endif
/* 162 */ NdrFcShort( 0x3da ),          /* Type Offset=986 */

          /* Return value */

/* 164 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
#ifndef _PPC_
#if !defined(_MIPS_)
/* 166 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
#else
                               NdrFcShort( 0x1c ), /*  MIPS Stack size/offset = 28 */
#endif
#else
                               NdrFcShort( 0x1c ), /* PPC Stack size/offset = 28 */
#endif
#else
                               NdrFcShort( 0x20 ), /* Alpha Stack size/offset = 32 */
#endif
/* 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 */
#ifndef _ALPHA_
/* 178 */ NdrFcShort( 0x8 ),   /* x86, MIPS, PPC Stack size/offset = 8 */
#else
                               NdrFcShort( 0x10 ), /* Alpha Stack size/offset = 16 */
#endif
/* 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, */
#ifndef _ALPHA_
/* 188 */ NdrFcShort( 0x4 ),   /* x86, MIPS, PPC Stack size/offset = 4 */
#else
                               NdrFcShort( 0x8 ),  /* Alpha Stack size/offset = 8 */
#endif
/* 190 */ 0x8,                 /* FC_LONG */
                               0x0,                /* 0 */

                               0x0
        }
    };

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                               NdrFcShort( 0x0 ),  /* 0 */
/*  2 */
                               0x12, 0x0,          /* FC_UP */
/*  4 */  NdrFcShort( 0x3b0 ),          /* Offset= 944 (948) */
/*  6 */
                               0x2b,               /* FC_NON_ENCAPSULATED_UNION */
```

```
                            0x9,                /* FC_ULONG */
/*  8 */  0x7,                /* Corr desc: FC_USHORT */
                            0x0,                /*  */
/* 10 */  NdrFcShort( 0xfff8 ),         /* -8 */
/* 12 */  NdrFcShort( 0x2 ),  /* Offset= 2 (14) */
/* 14 */  NdrFcShort( 0x10 ), /* 16 */
/* 16 */  NdrFcShort( 0x2b ), /* 43 */
/* 18 */  NdrFcLong( 0x3 ),   /* 3 */
/* 22 */  NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 24 */  NdrFcLong( 0x11 ),  /* 17 */
/* 28 */  NdrFcShort( 0x8001 ),         /* Simple arm type: FC_BYTE */
/* 30 */  NdrFcLong( 0x2 ),   /* 2 */
/* 34 */  NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 36 */  NdrFcLong( 0x4 ),   /* 4 */
/* 40 */  NdrFcShort( 0x800a ),         /* Simple arm type: FC_FLOAT */
/* 42 */  NdrFcLong( 0x5 ),   /* 5 */
/* 46 */  NdrFcShort( 0x800c ),         /* Simple arm type: FC_DOUBLE */
/* 48 */  NdrFcLong( 0xb ),   /* 11 */
/* 52 */  NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 54 */  NdrFcLong( 0xa ),   /* 10 */
/* 58 */  NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 60 */  NdrFcLong( 0x6 ),   /* 6 */
/* 64 */  NdrFcShort( 0xd6 ), /* Offset= 214 (278) */
/* 66 */  NdrFcLong( 0x7 ),   /* 7 */
/* 70 */  NdrFcShort( 0x800c ),         /* Simple arm type: FC_DOUBLE */
/* 72 */  NdrFcLong( 0x8 ),   /* 8 */
/* 76 */  NdrFcShort( 0xd0 ), /* Offset= 208 (284) */
/* 78 */  NdrFcLong( 0xd ),   /* 13 */
/* 82 */  NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */  NdrFcLong( 0x9 ),   /* 9 */
/* 88 */  NdrFcShort( 0xee ), /* Offset= 238 (326) */
/* 90 */  NdrFcLong( 0x2000 ),          /* 8192 */
/* 94 */  NdrFcShort( 0xfa ), /* Offset= 250 (344) */
/* 96 */  NdrFcLong( 0x24 ),  /* 36 */
/* 100 */ NdrFcShort( 0x308 ),          /* Offset= 776 (876) */
/* 102 */ NdrFcLong( 0x4024 ),          /* 16420 */
/* 106 */ NdrFcShort( 0x302 ),          /* Offset= 770 (876) */
/* 108 */ NdrFcLong( 0x4011 ),          /* 16401 */
/* 112 */ NdrFcShort( 0x300 ),          /* Offset= 768 (880) */
/* 114 */ NdrFcLong( 0x4002 ),          /* 16386 */
/* 118 */ NdrFcShort( 0x2fe ),          /* Offset= 766 (884) */
/* 120 */ NdrFcLong( 0x4003 ),          /* 16387 */
/* 124 */ NdrFcShort( 0x2fc ),          /* Offset= 764 (888) */
/* 126 */ NdrFcLong( 0x4004 ),          /* 16388 */
/* 130 */ NdrFcShort( 0x2fa ),          /* Offset= 762 (892) */
/* 132 */ NdrFcLong( 0x4005 ),          /* 16389 */
/* 136 */ NdrFcShort( 0x2f8 ),          /* Offset= 760 (896) */
/* 138 */ NdrFcLong( 0x400b ),          /* 16395 */
/* 142 */ NdrFcShort( 0x2e6 ),          /* Offset= 742 (884) */
/* 144 */ NdrFcLong( 0x400a ),          /* 16394 */
/* 148 */ NdrFcShort( 0x2e4 ),          /* Offset= 740 (888) */
/* 150 */ NdrFcLong( 0x4006 ),          /* 16390 */
/* 154 */ NdrFcShort( 0x2ea ),          /* Offset= 746 (900) */
/* 156 */ NdrFcLong( 0x4007 ),          /* 16391 */
/* 160 */ NdrFcShort( 0x2e0 ),          /* Offset= 736 (896) */
/* 162 */ NdrFcLong( 0x4008 ),          /* 16392 */
/* 166 */ NdrFcShort( 0x2e2 ),          /* Offset= 738 (904) */
/* 168 */ NdrFcLong( 0x400d ),          /* 16397 */
/* 172 */ NdrFcShort( 0x2e0 ),          /* Offset= 736 (908) */
/* 174 */ NdrFcLong( 0x4009 ),          /* 16393 */
/* 178 */ NdrFcShort( 0x2de ),          /* Offset= 734 (912) */
/* 180 */ NdrFcLong( 0x6000 ),          /* 24576 */
/* 184 */ NdrFcShort( 0x2dc ),          /* Offset= 732 (916) */
/* 186 */ NdrFcLong( 0x400c ),          /* 16396 */
/* 190 */ NdrFcShort( 0x2da ),          /* Offset= 730 (920) */
/* 192 */ NdrFcLong( 0x10 ),  /* 16 */
/* 196 */ NdrFcShort( 0x8002 ),         /* Simple arm type: FC_CHAR */
/* 198 */ NdrFcLong( 0x12 ),  /* 18 */
/* 202 */ NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 204 */ NdrFcLong( 0x13 ),  /* 19 */
/* 208 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 210 */ NdrFcLong( 0x16 ),  /* 22 */
/* 214 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 216 */ NdrFcLong( 0x17 ),  /* 23 */
/* 220 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 222 */ NdrFcLong( 0xe ),   /* 14 */
/* 226 */ NdrFcShort( 0x2be ),          /* Offset= 702 (928) */
/* 228 */ NdrFcLong( 0x400e ),          /* 16398 */
/* 232 */ NdrFcShort( 0x2c4 ),          /* Offset= 708 (940) */
/* 234 */ NdrFcLong( 0x4010 ),          /* 16400 */
/* 238 */ NdrFcShort( 0x2c2 ),          /* Offset= 706 (944) */
/* 240 */ NdrFcLong( 0x4012 ),          /* 16402 */
/* 244 */ NdrFcShort( 0x280 ),          /* Offset= 640 (884) */
/* 246 */ NdrFcLong( 0x4013 ),          /* 16403 */
/* 250 */ NdrFcShort( 0x27e ),          /* Offset= 638 (888) */
/* 252 */ NdrFcLong( 0x4016 ),          /* 16406 */
/* 256 */ NdrFcShort( 0x278 ),          /* Offset= 632 (888) */
/* 258 */ NdrFcLong( 0x4017 ),          /* 16407 */
/* 262 */ NdrFcShort( 0x272 ),          /* Offset= 626 (888) */
/* 264 */ NdrFcLong( 0x0 ),   /* 0 */
/* 268 */ NdrFcShort( 0x0 ),  /* Offset= 0 (268) */
/* 270 */ NdrFcLong( 0x1 ),   /* 1 */
/* 274 */ NdrFcShort( 0x0 ),  /* Offset= 0 (274) */
```

```
/* 276 */ NdrFcShort( 0xffffffff ),     /* Offset= -1 (275) */
/* 278 */
                           0x15,               /* FC_STRUCT */
                           0x7,                /* 7 */
/* 280 */ NdrFcShort( 0x8 ),  /* 8 */
/* 282 */ 0xb,                /* FC_HYPER */
                           0x5b,               /* FC_END */
/* 284 */
                           0x12, 0x0,          /* FC_UP */
/* 286 */ NdrFcShort( 0xc ),  /* Offset= 12 (298) */
/* 288 */
                           0x1b,               /* FC_CARRAY */
                           0x1,                /* 1 */
/* 290 */ NdrFcShort( 0x2 ),  /* 2 */
/* 292 */ 0x9,                /* Corr desc: FC_ULONG */
                           0x0,                /*  */
/* 294 */ NdrFcShort( 0xfffc ),         /* -4 */
/* 296 */ 0x6,                /* FC_SHORT */
                           0x5b,               /* FC_END */
/* 298 */
                           0x17,               /* FC_CSTRUCT */
                           0x3,                /* 3 */
/* 300 */ NdrFcShort( 0x8 ),  /* 8 */
/* 302 */ NdrFcShort( 0xfffffff2 ),     /* Offset= -14 (288) */
/* 304 */ 0x8,                /* FC_LONG */
                           0x8,                /* FC_LONG */
/* 306 */ 0x5c,               /* FC_PAD */
                           0x5b,               /* FC_END */
/* 308 */
                           0x2f,               /* FC_IP */
                           0x5a,               /* FC_CONSTANT_IID */
/* 310 */ NdrFcLong( 0x0 ),   /* 0 */
/* 314 */ NdrFcShort( 0x0 ),  /* 0 */
/* 316 */ NdrFcShort( 0x0 ),  /* 0 */
/* 318 */ 0xc0,               /* 192 */
                           0x0,                /* 0 */
/* 320 */ 0x0,                /* 0 */
                           0x0,                /* 0 */
/* 322 */ 0x0,                /* 0 */
                           0x0,                /* 0 */
/* 324 */ 0x0,                /* 0 */
                           0x46,               /* 70 */
/* 326 */
                           0x2f,               /* FC_IP */
                           0x5a,               /* FC_CONSTANT_IID */
/* 328 */ NdrFcLong( 0x20400 ),         /* 132096 */
/* 332 */ NdrFcShort( 0x0 ),  /* 0 */
/* 334 */ NdrFcShort( 0x0 ),  /* 0 */
/* 336 */ 0xc0,               /* 192 */
                           0x0,                /* 0 */
/* 338 */ 0x0,                /* 0 */
                           0x0,                /* 0 */
/* 340 */ 0x0,                /* 0 */
                           0x0,                /* 0 */
/* 342 */ 0x0,                /* 0 */
                           0x46,               /* 70 */
/* 344 */
                           0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 346 */ NdrFcShort( 0x2 ),  /* Offset= 2 (348) */
/* 348 */
                           0x12, 0x0,          /* FC_UP */
/* 350 */ NdrFcShort( 0x1fc ),          /* Offset= 508 (858) */
/* 352 */
                           0x2a,               /* FC_ENCAPSULATED_UNION */
                           0x49,               /* 73 */
/* 354 */ NdrFcShort( 0x18 ), /* 24 */
/* 356 */ NdrFcShort( 0xa ),  /* 10 */
/* 358 */ NdrFcLong( 0x8 ),   /* 8 */
/* 362 */ NdrFcShort( 0x58 ), /* Offset= 88 (450) */
/* 364 */ NdrFcLong( 0xd ),   /* 13 */
/* 368 */ NdrFcShort( 0x78 ), /* Offset= 120 (488) */
/* 370 */ NdrFcLong( 0x9 ),   /* 9 */
/* 374 */ NdrFcShort( 0x94 ), /* Offset= 148 (522) */
/* 376 */ NdrFcLong( 0xc ),   /* 12 */
/* 380 */ NdrFcShort( 0xbc ), /* Offset= 188 (568) */
/* 382 */ NdrFcLong( 0x24 ),  /* 36 */
/* 386 */ NdrFcShort( 0x114 ),          /* Offset= 276 (662) */
/* 388 */ NdrFcLong( 0x800d ),          /* 32781 */
/* 392 */ NdrFcShort( 0x130 ),          /* Offset= 304 (696) */
/* 394 */ NdrFcLong( 0x10 ),  /* 16 */
/* 398 */ NdrFcShort( 0x148 ),          /* Offset= 328 (726) */
/* 400 */ NdrFcLong( 0x2 ),   /* 2 */
/* 404 */ NdrFcShort( 0x160 ),          /* Offset= 352 (756) */
/* 406 */ NdrFcLong( 0x3 ),   /* 3 */
/* 410 */ NdrFcShort( 0x178 ),          /* Offset= 376 (786) */
/* 412 */ NdrFcLong( 0x14 ),  /* 20 */
/* 416 */ NdrFcShort( 0x190 ),          /* Offset= 400 (816) */
/* 418 */ NdrFcShort( 0xffffffff ),     /* Offset= -1 (417) */
/* 420 */
                           0x1b,               /* FC_CARRAY */
                           0x3,                /* 3 */
/* 422 */ NdrFcShort( 0x4 ),  /* 4 */
/* 424 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
```

```
                                0x0,                /*  */
/* 426 */ NdrFcShort( 0x0 ),    /* 0 */
/* 428 */
                                0x4b,               /* FC_PP */
                                0x5c,               /* FC_PAD */
/* 430 */
                                0x48,               /* FC_VARIABLE_REPEAT */
                                0x49,               /* FC_FIXED_OFFSET */
/* 432 */ NdrFcShort( 0x4 ),    /* 4 */
/* 434 */ NdrFcShort( 0x0 ),    /* 0 */
/* 436 */ NdrFcShort( 0x1 ),    /* 1 */
/* 438 */ NdrFcShort( 0x0 ),    /* 0 */
/* 440 */ NdrFcShort( 0x0 ),    /* 0 */
/* 442 */ 0x12, 0x0,            /* FC_UP */
/* 444 */ NdrFcShort( 0xffffff6e ),     /* Offset= -146 (298) */
/* 446 */
                                0x5b,               /* FC_END */

                                0x8,                /* FC_LONG */
/* 448 */ 0x5c,                 /* FC_PAD */
                                0x5b,               /* FC_END */
/* 450 */
                                0x16,               /* FC_PSTRUCT */
                                0x3,                /* 3 */
/* 452 */ NdrFcShort( 0x8 ),    /* 8 */
/* 454 */
                                0x4b,               /* FC_PP */
                                0x5c,               /* FC_PAD */
/* 456 */
                                0x46,               /* FC_NO_REPEAT */
                                0x5c,               /* FC_PAD */
/* 458 */ NdrFcShort( 0x4 ),    /* 4 */
/* 460 */ NdrFcShort( 0x4 ),    /* 4 */
/* 462 */ 0x11, 0x0,            /* FC_RP */
/* 464 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44 (420) */
/* 466 */
                                0x5b,               /* FC_END */

                                0x8,                /* FC_LONG */
/* 468 */ 0x8,                  /* FC_LONG */
                                0x5b,               /* FC_END */
/* 470 */
                                0x21,               /* FC_BOGUS_ARRAY */
                                0x3,                /* 3 */
/* 472 */ NdrFcShort( 0x0 ),    /* 0 */
/* 474 */ 0x19,                 /* Corr desc:  field pointer, FC_ULONG */
                                0x0,                /*  */
/* 476 */ NdrFcShort( 0x0 ),    /* 0 */
/* 478 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 482 */ 0x4c,                 /* FC_EMBEDDED_COMPLEX */
                                0x0,                /* 0 */
/* 484 */ NdrFcShort( 0xffffff50 ),     /* Offset= -176 (308) */
/* 486 */ 0x5c,                 /* FC_PAD */
                                0x5b,               /* FC_END */
/* 488 */
                                0x1a,               /* FC_BOGUS_STRUCT */
                                0x3,                /* 3 */
/* 490 */ NdrFcShort( 0x8 ),    /* 8 */
/* 492 */ NdrFcShort( 0x0 ),    /* 0 */
/* 494 */ NdrFcShort( 0x6 ),    /* Offset= 6 (500) */
/* 496 */ 0x8,                  /* FC_LONG */
                                0x36,               /* FC_POINTER */
/* 498 */ 0x5c,                 /* FC_PAD */
                                0x5b,               /* FC_END */
/* 500 */
                                0x11, 0x0,          /* FC_RP */
/* 502 */ NdrFcShort( 0xfffffe0 ),      /* Offset= -32 (470) */
/* 504 */
                                0x21,               /* FC_BOGUS_ARRAY */
                                0x3,                /* 3 */
/* 506 */ NdrFcShort( 0x0 ),    /* 0 */
/* 508 */ 0x19,                 /* Corr desc:  field pointer, FC_ULONG */
                                0x0,                /*  */
/* 510 */ NdrFcShort( 0x0 ),    /* 0 */
/* 512 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 516 */ 0x4c,                 /* FC_EMBEDDED_COMPLEX */
                                0x0,                /* 0 */
/* 518 */ NdrFcShort( 0xffffff40 ),     /* Offset= -192 (326) */
/* 520 */ 0x5c,                 /* FC_PAD */
                                0x5b,               /* FC_END */
/* 522 */
                                0x1a,               /* FC_BOGUS_STRUCT */
                                0x3,                /* 3 */
/* 524 */ NdrFcShort( 0x8 ),    /* 8 */
/* 526 */ NdrFcShort( 0x0 ),    /* 0 */
/* 528 */ NdrFcShort( 0x6 ),    /* Offset= 6 (534) */
/* 530 */ 0x8,                  /* FC_LONG */
                                0x36,               /* FC_POINTER */
/* 532 */ 0x5c,                 /* FC_PAD */
                                0x5b,               /* FC_END */
/* 534 */
                                0x11, 0x0,          /* FC_RP */
/* 536 */ NdrFcShort( 0xfffffe0 ),      /* Offset= -32 (504) */
```

```
/* 538 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 540 */ NdrFcShort( 0x4 ),  /* 4 */
/* 542 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 544 */ NdrFcShort( 0x0 ),  /* 0 */
/* 546 */
                            0x4b,               /* FC_PP */
                            0x5c,               /* FC_PAD */
/* 548 */
                            0x48,               /* FC_VARIABLE_REPEAT */
                            0x49,               /* FC_FIXED_OFFSET */
/* 550 */ NdrFcShort( 0x4 ),  /* 4 */
/* 552 */ NdrFcShort( 0x0 ),  /* 0 */
/* 554 */ NdrFcShort( 0x1 ),  /* 1 */
/* 556 */ NdrFcShort( 0x0 ),  /* 0 */
/* 558 */ NdrFcShort( 0x0 ),  /* 0 */
/* 560 */ 0x12, 0x0,          /* FC_UP */
/* 562 */ NdrFcShort( 0x182 ),          /* Offset= 386 (948) */
/* 564 */
                            0x5b,               /* FC_END */

                            0x8,                /* FC_LONG */
/* 566 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 568 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 570 */ NdrFcShort( 0x8 ),  /* 8 */
/* 572 */ NdrFcShort( 0x0 ),  /* 0 */
/* 574 */ NdrFcShort( 0x6 ),  /* Offset= 6 (580) */
/* 576 */ 0x8,                /* FC_LONG */
                            0x36,               /* FC_POINTER */
/* 578 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 580 */
                            0x11, 0x0,          /* FC_RP */
/* 582 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44 (538) */
/* 584 */
                            0x2f,               /* FC_IP */
                            0x5a,               /* FC_CONSTANT_IID */
/* 586 */ NdrFcLong( 0x2f ),  /* 47 */
/* 590 */ NdrFcShort( 0x0 ),  /* 0 */
/* 592 */ NdrFcShort( 0x0 ),  /* 0 */
/* 594 */ 0xc0,               /* 192 */
                            0x0,                /* 0 */
/* 596 */ 0x0,                /* 0 */
                            0x0,                /* 0 */
/* 598 */ 0x0,                /* 0 */
                            0x0,                /* 0 */
/* 600 */ 0x0,                /* 0 */
                            0x46,               /* 70 */
/* 602 */
                            0x1b,               /* FC_CARRAY */
                            0x0,                /* 0 */
/* 604 */ NdrFcShort( 0x1 ),  /* 1 */
/* 606 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 608 */ NdrFcShort( 0x4 ),  /* 4 */
/* 610 */ 0x1,                /* FC_BYTE */
                            0x5b,               /* FC_END */
/* 612 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 614 */ NdrFcShort( 0x10 ), /* 16 */
/* 616 */ NdrFcShort( 0x0 ),  /* 0 */
/* 618 */ NdrFcShort( 0xa ),  /* Offset= 10 (628) */
/* 620 */ 0x8,                /* FC_LONG */
                            0x8,                /* FC_LONG */
/* 622 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                            0x0,                /* 0 */
/* 624 */ NdrFcShort( 0xffffffd8 ),     /* Offset= -40 (584) */
/* 626 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 628 */
                            0x12, 0x0,          /* FC_UP */
/* 630 */ NdrFcShort( 0xfffffffe4 ),     /* Offset= -28 (602) */
/* 632 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 634 */ NdrFcShort( 0x4 ),  /* 4 */
/* 636 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 638 */ NdrFcShort( 0x0 ),  /* 0 */
/* 640 */
                            0x4b,               /* FC_PP */
                            0x5c,               /* FC_PAD */
/* 642 */
                            0x48,               /* FC_VARIABLE_REPEAT */
                            0x49,               /* FC_FIXED_OFFSET */
/* 644 */ NdrFcShort( 0x4 ),  /* 4 */
/* 646 */ NdrFcShort( 0x0 ),  /* 0 */
```

```
/* 648 */ NdrFcShort( 0x1 ),  /* 1 */
/* 650 */ NdrFcShort( 0x0 ),  /* 0 */
/* 652 */ NdrFcShort( 0x0 ),  /* 0 */
/* 654 */ 0x12, 0x0,          /* FC_UP */
/* 656 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44 (612) */
/* 658 */
                               0x5b,               /* FC_END */

                               0x8,                /* FC_LONG */
/* 660 */ 0x5c,                /* FC_PAD */
                               0x5b,               /* FC_END */
/* 662 */
                               0x1a,               /* FC_BOGUS_STRUCT */
                               0x3,                /* 3 */
/* 664 */ NdrFcShort( 0x8 ),  /* 8 */
/* 666 */ NdrFcShort( 0x0 ),  /* 0 */
/* 668 */ NdrFcShort( 0x6 ),  /* Offset= 6 (674) */
/* 670 */ 0x8,                /* FC_LONG */
                               0x36,               /* FC_POINTER */
/* 672 */ 0x5c,                /* FC_PAD */
                               0x5b,               /* FC_END */
/* 674 */
                               0x11, 0x0,          /* FC_RP */
/* 676 */ NdrFcShort( 0xffffffd4 ),     /* Offset= -44 (632) */
/* 678 */
                               0x1d,               /* FC_SMFARRAY */
                               0x0,                /* 0 */
/* 680 */ NdrFcShort( 0x8 ),  /* 8 */
/* 682 */ 0x2,                /* FC_CHAR */
                               0x5b,               /* FC_END */
/* 684 */
                               0x15,               /* FC_STRUCT */
                               0x3,                /* 3 */
/* 686 */ NdrFcShort( 0x10 ), /* 16 */
/* 688 */ 0x8,                /* FC_LONG */
                               0x6,                /* FC_SHORT */
/* 690 */ 0x6,                /* FC_SHORT */
                               0x4c,               /* FC_EMBEDDED_COMPLEX */
/* 692 */ 0x0,                /* 0 */
                               NdrFcShort( 0xfffffff1 ),     /* Offset= -15 (678) */
                               0x5b,               /* FC_END */
/* 696 */
                               0x1a,               /* FC_BOGUS_STRUCT */
                               0x3,                /* 3 */
/* 698 */ NdrFcShort( 0x18 ), /* 24 */
/* 700 */ NdrFcShort( 0x0 ),  /* 0 */
/* 702 */ NdrFcShort( 0xa ),  /* Offset= 10 (712) */
/* 704 */ 0x8,                /* FC_LONG */
                               0x36,               /* FC_POINTER */
/* 706 */ 0x4c,                /* FC_EMBEDDED_COMPLEX */
                               0x0,                /* 0 */
/* 708 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (684) */
/* 710 */ 0x5c,                /* FC_PAD */
                               0x5b,               /* FC_END */
/* 712 */
                               0x11, 0x0,          /* FC_RP */
/* 714 */ NdrFcShort( 0xffffff0c ),     /* Offset= -244 (470) */
/* 716 */
                               0x1b,               /* FC_CARRAY */
                               0x0,                /* 0 */
/* 718 */ NdrFcShort( 0x1 ),  /* 1 */
/* 720 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                               0x0,                /*  */
/* 722 */ NdrFcShort( 0x0 ),  /* 0 */
/* 724 */ 0x1,                /* FC_BYTE */
                               0x5b,               /* FC_END */
/* 726 */
                               0x16,               /* FC_PSTRUCT */
                               0x3,                /* 3 */
/* 728 */ NdrFcShort( 0x8 ),  /* 8 */
/* 730 */
                               0x4b,               /* FC_PP */
                               0x5c,               /* FC_PAD */
/* 732 */
                               0x46,               /* FC_NO_REPEAT */
                               0x5c,               /* FC_PAD */
/* 734 */ NdrFcShort( 0x4 ),  /* 4 */
/* 736 */ NdrFcShort( 0x4 ),  /* 4 */
/* 738 */ 0x12, 0x0,          /* FC_UP */
/* 740 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (716) */
/* 742 */
                               0x5b,               /* FC_END */

                               0x8,                /* FC_LONG */
/* 744 */ 0x8,                /* FC_LONG */
                               0x5b,               /* FC_END */
/* 746 */
                               0x1b,               /* FC_CARRAY */
                               0x1,                /* 1 */
/* 748 */ NdrFcShort( 0x2 ),  /* 2 */
/* 750 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                               0x0,                /*  */
/* 752 */ NdrFcShort( 0x0 ),  /* 0 */
```

```
/* 754 */ 0x6,              /* FC_SHORT */
                            0x5b,               /* FC_END */
/* 756 */
                            0x16,               /* FC_PSTRUCT */
                            0x3,                /* 3 */
/* 758 */ NdrFcShort( 0x8 ),  /* 8 */
/* 760 */
                            0x4b,               /* FC_PP */
                            0x5c,               /* FC_PAD */
/* 762 */
                            0x46,               /* FC_NO_REPEAT */
                            0x5c,               /* FC_PAD */
/* 764 */ NdrFcShort( 0x4 ),  /* 4 */
/* 766 */ NdrFcShort( 0x4 ),  /* 4 */
/* 768 */ 0x12, 0x0,          /* FC_UP */
/* 770 */ NdrFcShort( 0xffffffe8 ),     /* Offset= -24 (746) */
/* 772 */
                            0x5b,               /* FC_END */

                            0x8,                /* FC_LONG */
/* 774 */ 0x8,              /* FC_LONG */
                            0x5b,               /* FC_END */
/* 776 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 778 */ NdrFcShort( 0x4 ),  /* 4 */
/* 780 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 782 */ NdrFcShort( 0x0 ),  /* 0 */
/* 784 */ 0x8,                /* FC_LONG */
                            0x5b,               /* FC_END */
/* 786 */
                            0x16,               /* FC_PSTRUCT */
                            0x3,                /* 3 */
/* 788 */ NdrFcShort( 0x8 ),  /* 8 */
/* 790 */
                            0x4b,               /* FC_PP */
                            0x5c,               /* FC_PAD */
/* 792 */
                            0x46,               /* FC_NO_REPEAT */
                            0x5c,               /* FC_PAD */
/* 794 */ NdrFcShort( 0x4 ),  /* 4 */
/* 796 */ NdrFcShort( 0x4 ),  /* 4 */
/* 798 */ 0x12, 0x0,          /* FC_UP */
/* 800 */ NdrFcShort( 0xfffffe8 ),     /* Offset= -24 (776) */
/* 802 */
                            0x5b,               /* FC_END */

                            0x8,                /* FC_LONG */
/* 804 */ 0x8,              /* FC_LONG */
                            0x5b,               /* FC_END */
/* 806 */
                            0x1b,               /* FC_CARRAY */
                            0x7,                /* 7 */
/* 808 */ NdrFcShort( 0x8 ),  /* 8 */
/* 810 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 812 */ NdrFcShort( 0x0 ),  /* 0 */
/* 814 */ 0xb,                /* FC_HYPER */
                            0x5b,               /* FC_END */
/* 816 */
                            0x16,               /* FC_PSTRUCT */
                            0x3,                /* 3 */
/* 818 */ NdrFcShort( 0x8 ),  /* 8 */
/* 820 */
                            0x4b,               /* FC_PP */
                            0x5c,               /* FC_PAD */
/* 822 */
                            0x46,               /* FC_NO_REPEAT */
                            0x5c,               /* FC_PAD */
/* 824 */ NdrFcShort( 0x4 ),  /* 4 */
/* 826 */ NdrFcShort( 0x4 ),  /* 4 */
/* 828 */ 0x12, 0x0,          /* FC_UP */
/* 830 */ NdrFcShort( 0xfffffe8 ),     /* Offset= -24 (806) */
/* 832 */
                            0x5b,               /* FC_END */

                            0x8,                /* FC_LONG */
/* 834 */ 0x8,              /* FC_LONG */
                            0x5b,               /* FC_END */
/* 836 */
                            0x15,               /* FC_STRUCT */
                            0x3,                /* 3 */
/* 838 */ NdrFcShort( 0x8 ),  /* 8 */
/* 840 */ 0x8,                /* FC_LONG */
                            0x8,                /* FC_LONG */
/* 842 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 844 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 846 */ NdrFcShort( 0x8 ),  /* 8 */
/* 848 */ 0x7,                /* Corr desc: FC_USHORT */
```

```
                            0x0,                /*  */
/* 850 */ NdrFcShort( 0xffd8 ),         /* -40 */
/* 852 */ 0x4c,                /* FC_EMBEDDED_COMPLEX */
                            0x0,                /* 0 */
/* 854 */ NdrFcShort( 0xfffffffe ),     /* Offset= -18 (836) */
/* 856 */ 0x5c,                /* FC_PAD */
                            0x5b,               /* FC_END */
/* 858 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 860 */ NdrFcShort( 0x28 ), /* 40 */
/* 862 */ NdrFcShort( 0xfffffffe ),     /* Offset= -18 (844) */
/* 864 */ NdrFcShort( 0x0 ),  /* Offset= 0 (864) */
/* 866 */ 0x6,                /* FC_SHORT */
                            0x6,                /* FC_SHORT */
/* 868 */ 0x38,               /* FC_ALIGNM4 */
                            0x8,                /* FC_LONG */
/* 870 */ 0x8,                /* FC_LONG */
                            0x4c,               /* FC_EMBEDDED_COMPLEX */
/* 872 */ 0x0,                /* 0 */
                            NdrFcShort( 0xfffffdf7 ),     /* Offset= -521 (352) */
                            0x5b,               /* FC_END */
/* 876 */
                            0x12, 0x0,          /* FC_UP */
/* 878 */ NdrFcShort( 0xfffffef6 ),     /* Offset= -266 (612) */
/* 880 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 882 */ 0x1,                /* FC_BYTE */
                            0x5c,               /* FC_PAD */
/* 884 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 886 */ 0x6,                /* FC_SHORT */
                            0x5c,               /* FC_PAD */
/* 888 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 890 */ 0x8,                /* FC_LONG */
                            0x5c,               /* FC_PAD */
/* 892 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 894 */ 0xa,                /* FC_FLOAT */
                            0x5c,               /* FC_PAD */
/* 896 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 898 */ 0xc,                /* FC_DOUBLE */
                            0x5c,               /* FC_PAD */
/* 900 */
                            0x12, 0x0,          /* FC_UP */
/* 902 */ NdrFcShort( 0xfffffd90 ),     /* Offset= -624 (278) */
/* 904 */
                            0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 906 */ NdrFcShort( 0xfffffd92 ),     /* Offset= -622 (284) */
/* 908 */
                            0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 910 */ NdrFcShort( 0xfffffda6 ),     /* Offset= -602 (308) */
/* 912 */
                            0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 914 */ NdrFcShort( 0xfffffdb4 ),     /* Offset= -588 (326) */
/* 916 */
                            0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 918 */ NdrFcShort( 0xfffffdc2 ),     /* Offset= -574 (344) */
/* 920 */
                            0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 922 */ NdrFcShort( 0x2 ),  /* Offset= 2 (924) */
/* 924 */
                            0x12, 0x0,          /* FC_UP */
/* 926 */ NdrFcShort( 0x16 ), /* Offset= 22 (948) */
/* 928 */
                            0x15,               /* FC_STRUCT */
                            0x7,                /* 7 */
/* 930 */ NdrFcShort( 0x10 ), /* 16 */
/* 932 */ 0x6,                /* FC_SHORT */
                            0x1,                /* FC_BYTE */
/* 934 */ 0x1,                /* FC_BYTE */
                            0x38,               /* FC_ALIGNM4 */
/* 936 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 938 */ 0xb,                /* FC_HYPER */
                            0x5b,               /* FC_END */
/* 940 */
                            0x12, 0x0,          /* FC_UP */
/* 942 */ NdrFcShort( 0xfffffff2 ),     /* Offset= -14 (928) */
/* 944 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 946 */ 0x2,                /* FC_CHAR */
                            0x5c,               /* FC_PAD */
/* 948 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x7,                /* 7 */
/* 950 */ NdrFcShort( 0x20 ), /* 32 */
/* 952 */ NdrFcShort( 0x0 ),  /* 0 */
/* 954 */ NdrFcShort( 0x0 ),  /* Offset= 0 (954) */
/* 956 */ 0x8,                /* FC_LONG */
                            0x8,                /* FC_LONG */
```

```
/* 958 */ 0x6,               /* FC_SHORT */
                              0x6,                /* FC_SHORT */
/* 960 */ 0x6,               /* FC_SHORT */
                              0x6,                /* FC_SHORT */
/* 962 */ 0x4c,              /* FC_EMBEDDED_COMPLEX */
                              0x0,                /* 0 */
/* 964 */ NdrFcShort( 0xfffffc42 ),     /* Offset= -958 (6) */
/* 966 */ 0x5c,              /* FC_PAD */
                              0x5b,               /* FC_END */
/* 968 */ 0xb4,              /* FC_USER_MARSHAL */
                              0x83,               /* 131 */
/* 970 */ NdrFcShort( 0x0 ),  /* 0 */
/* 972 */ NdrFcShort( 0x10 ), /* 16 */
/* 974 */ NdrFcShort( 0x0 ),  /* 0 */
/* 976 */ NdrFcShort( 0xfffffc32 ),     /* Offset= -974 (2) */
/* 978 */
                              0x11, 0x4,          /* FC_RP [alloced_on_stack] */
/* 980 */ NdrFcShort( 0x6 ),  /* Offset= 6 (986) */
/* 982 */
                              0x13, 0x0,          /* FC_OP */
/* 984 */ NdrFcShort( 0xffffffdc ),     /* Offset= -36 (948) */
/* 986 */ 0xb4,              /* FC_USER_MARSHAL */
                              0x83,               /* 131 */
/* 988 */ NdrFcShort( 0x0 ),  /* 0 */
/* 990 */ NdrFcShort( 0x10 ), /* 16 */
/* 992 */ NdrFcShort( 0x0 ),  /* 0 */
/* 994 */ NdrFcShort( 0xfffffff4 ),     /* Offset= -12 (982) */

                              0x0
        }
    };

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

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

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


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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

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


#endif /* !defined(_M_IA64) && !defined(_M_AXP64)*/


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

/* this ALWAYS GENERATED file contains the proxy stub code */


 /* File created by MIDL compiler version 5.03.0280 */
/* at Sat Apr 08 16:40:10 2000
 */
/* Compiler settings for .\src\tpcc_com_ps.idl:
```

```
    Oicf (OptLev=i2), W1, Zp8, env=Win64 (32b run,appending), 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_AXP64)
#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   979
#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;


extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;


/* 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}} */


extern const MIDL_STUB_DESC Object_StubDesc;


extern const MIDL_SERVER_INFO ITPCC_ServerInfo;

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
    0,
    44,
    88,
    132,
    176,
    220
    };

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

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

```
    0,
    0,
    0
    };

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

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

extern const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

static const MIDL_STUB_DESC Object_StubDesc =
    {
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x5030118, /* MIDL Version 5.4.280 */
    0,
    UserMarshalRoutines,
    0,  /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0,  /* Reserved3 */
    0,  /* Reserved4 */
    0   /* Reserved5 */
    };

#pragma data_seg(".rdata")

static const USER_MARSHAL_ROUTINE_QUADRUPLE UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
        {

            {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
            }

        };


#if !defined(__RPC_WIN64__)
#error  Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
    {
        0,
        {

          /* Procedure NewOrder */

                               0x33,               /* FC_AUTO_HANDLE */
                               0x6c,               /* Old Flags:  object, Oi2 */
/*  2 */  NdrFcLong( 0x0 ),   /* 0 */
/*  6 */  NdrFcShort( 0x3 ),  /* 3 */
#ifndef _ALPHA_
/*  8 */  NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                               NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 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, */
#ifndef _ALPHA_
/* 28 */  NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ),  /* axp64 Stack size/offset = 8 */
#endif
/* 30 */  NdrFcShort( 0x3b6 ),          /* Type Offset=950 */

          /* Parameter txn_out */

/* 32 */  NdrFcShort( 0x6113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 34 */  NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 36 */  NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Return value */

/* 38 */  NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 40 */  NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 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 */
#ifndef _ALPHA_
/* 52 */  NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                                NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 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, */
#ifndef _ALPHA_
/* 72 */  NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                                NdrFcShort( 0x8 ),  /* axp64 Stack size/offset = 8 */
#endif
/* 74 */  NdrFcShort( 0x3b6 ),          /* Type Offset=950 */

          /* Parameter txn_out */

/* 76 */  NdrFcShort( 0x6113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 78 */  NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                                NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 80 */  NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Return value */

/* 82 */  NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 84 */  NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                                NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 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 */
#ifndef _ALPHA_
/* 96 */  NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                               NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 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, */
#ifndef _ALPHA_
/* 116 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                               NdrFcShort( 0x8 ),  /* axp64 Stack size/offset = 8 */
#endif
/* 118 */ NdrFcShort( 0x3b6 ),          /* Type Offset=950 */

          /* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 122 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                               NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 124 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

          /* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 128 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                               NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 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 */
#ifndef _ALPHA_
/* 140 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                               NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 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, */
#ifndef _ALPHA_
/* 160 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                               NdrFcShort( 0x8 ),  /* axp64 Stack size/offset = 8 */
#endif
/* 162 */ NdrFcShort( 0x3b6 ),          /* Type Offset=950 */

          /* Parameter txn_out */

/* 164 */ NdrFcShort( 0x6113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 166 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                               NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 168 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */
```

```
         /* Return value */

/* 170 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 172 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                              NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 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 */
#ifndef _ALPHA_
/* 184 */ NdrFcShort( 0x38 ), /* ia64 Stack size/offset = 56 */
#else
                              NdrFcShort( 0x30 ), /* axp64 Stack size/offset = 48 */
#endif
/* 186 */ NdrFcShort( 0x0 ),  /* 0 */
/* 188 */ NdrFcShort( 0x8 ),  /* 8 */
/* 190 */ 0x47,               /* Oi2 Flags:  srv must size, clt must size, has return, has ext, */
                              0x3,                /* 3 */
/* 192 */ 0xa,                /* 10 */
                              0x7,                /* Ext Flags:  new corr desc, clt corr check, srv corr check, */
/* 194 */ NdrFcShort( 0x20 ), /* 32 */
/* 196 */ NdrFcShort( 0x20 ), /* 32 */
/* 198 */ NdrFcShort( 0x0 ),  /* 0 */
/* 200 */ NdrFcShort( 0x0 ),  /* 0 */

         /* Parameter txn_in */

/* 202 */ NdrFcShort( 0x8b ), /* Flags:  must size, must free, in, by val, */
#ifndef _ALPHA_
/* 204 */ NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
#else
                              NdrFcShort( 0x8 ),  /* axp64 Stack size/offset = 8 */
#endif
/* 206 */ NdrFcShort( 0x3b6 ),          /* Type Offset=950 */

         /* Parameter txn_out */

/* 208 */ NdrFcShort( 0x6113 ),         /* Flags:  must size, must free, out, simple ref, srv alloc size=24 */
#ifndef _ALPHA_
/* 210 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
#else
                              NdrFcShort( 0x20 ), /* axp64 Stack size/offset = 32 */
#endif
/* 212 */ NdrFcShort( 0x3c8 ),          /* Type Offset=968 */

         /* Return value */

/* 214 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
#ifndef _ALPHA_
/* 216 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
#else
                              NdrFcShort( 0x28 ), /* axp64 Stack size/offset = 40 */
#endif
/* 218 */ 0x8,                /* FC_LONG */
                              0x0,                /* 0 */

         /* Procedure CallSetComplete */

/* 220 */ 0x33,               /* FC_AUTO_HANDLE */
                              0x6c,               /* Old Flags:  object, Oi2 */
/* 222 */ NdrFcLong( 0x0 ),   /* 0 */
/* 226 */ NdrFcShort( 0x8 ),  /* 8 */
/* 228 */ NdrFcShort( 0x10 ), /* ia64, axp64 Stack size/offset = 16 */
/* 230 */ NdrFcShort( 0x0 ),  /* 0 */
/* 232 */ NdrFcShort( 0x8 ),  /* 8 */
/* 234 */ 0x44,               /* Oi2 Flags:  has return, has ext, */
                              0x1,                /* 1 */
/* 236 */ 0xa,                /* 10 */
                              0x1,                /* Ext Flags:  new corr desc, */
/* 238 */ NdrFcShort( 0x0 ),  /* 0 */
/* 240 */ NdrFcShort( 0x0 ),  /* 0 */
/* 242 */ NdrFcShort( 0x0 ),  /* 0 */
/* 244 */ NdrFcShort( 0x0 ),  /* 0 */

         /* Return value */

/* 246 */ NdrFcShort( 0x70 ), /* Flags:  out, return, base type, */
/* 248 */ NdrFcShort( 0x8 ),  /* ia64, axp64 Stack size/offset = 8 */
/* 250 */ 0x8,                /* FC_LONG */
                              0x0,                /* 0 */

                              0x0
       }
   };
```

```
static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
    {
        0,
        {
                                NdrFcShort( 0x0 ),  /* 0 */
/*  2 */
                                0x12, 0x0,          /* FC_UP */
/*  4 */  NdrFcShort( 0x39e ),          /* Offset= 926 (930) */
/*  6 */
                                0x2b,               /* FC_NON_ENCAPSULATED_UNION */
                                0x9,                /* FC_ULONG */
/*  8 */  0x7,                /* Corr desc: FC_USHORT */
                                0x0,                /*  */
/* 10 */  NdrFcShort( 0xfff8 ),         /* -8 */
/* 12 */  NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 14 */  NdrFcShort( 0x2 ),  /* Offset= 2 (16) */
/* 16 */  NdrFcShort( 0x10 ), /* 16 */
/* 18 */  NdrFcShort( 0x2b ), /* 43 */
/* 20 */  NdrFcLong( 0x3 ),   /* 3 */
/* 24 */  NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 26 */  NdrFcLong( 0x11 ),  /* 17 */
/* 30 */  NdrFcShort( 0x8001 ),         /* Simple arm type: FC_BYTE */
/* 32 */  NdrFcLong( 0x2 ),   /* 2 */
/* 36 */  NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 38 */  NdrFcLong( 0x4 ),   /* 4 */
/* 42 */  NdrFcShort( 0x800a ),         /* Simple arm type: FC_FLOAT */
/* 44 */  NdrFcLong( 0x5 ),   /* 5 */
/* 48 */  NdrFcShort( 0x800c ),         /* Simple arm type: FC_DOUBLE */
/* 50 */  NdrFcLong( 0xb ),   /* 11 */
/* 54 */  NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 56 */  NdrFcLong( 0xa ),   /* 10 */
/* 60 */  NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 62 */  NdrFcLong( 0x6 ),   /* 6 */
/* 66 */  NdrFcShort( 0xd6 ), /* Offset= 214 (280) */
/* 68 */  NdrFcLong( 0x7 ),   /* 7 */
/* 72 */  NdrFcShort( 0x800c ),         /* Simple arm type: FC_DOUBLE */
/* 74 */  NdrFcLong( 0x8 ),   /* 8 */
/* 78 */  NdrFcShort( 0xd0 ), /* Offset= 208 (286) */
/* 80 */  NdrFcLong( 0xd ),   /* 13 */
/* 84 */  NdrFcShort( 0xe4 ), /* Offset= 228 (312) */
/* 86 */  NdrFcLong( 0x9 ),   /* 9 */
/* 90 */  NdrFcShort( 0xf0 ), /* Offset= 240 (330) */
/* 92 */  NdrFcLong( 0x2000 ),          /* 8192 */
/* 96 */  NdrFcShort( 0xfc ), /* Offset= 252 (348) */
/* 98 */  NdrFcLong( 0x24 ),  /* 36 */
/* 102 */ NdrFcShort( 0x2f4 ),          /* Offset= 756 (858) */
/* 104 */ NdrFcLong( 0x4024 ),          /* 16420 */
/* 108 */ NdrFcShort( 0x2ee ),          /* Offset= 750 (858) */
/* 110 */ NdrFcLong( 0x4011 ),          /* 16401 */
/* 114 */ NdrFcShort( 0x2ec ),          /* Offset= 748 (862) */
/* 116 */ NdrFcLong( 0x4002 ),          /* 16386 */
/* 120 */ NdrFcShort( 0x2ea ),          /* Offset= 746 (866) */
/* 122 */ NdrFcLong( 0x4003 ),          /* 16387 */
/* 126 */ NdrFcShort( 0x2e8 ),          /* Offset= 744 (870) */
/* 128 */ NdrFcLong( 0x4004 ),          /* 16388 */
/* 132 */ NdrFcShort( 0x2e6 ),          /* Offset= 742 (874) */
/* 134 */ NdrFcLong( 0x4005 ),          /* 16389 */
/* 138 */ NdrFcShort( 0x2e4 ),          /* Offset= 740 (878) */
/* 140 */ NdrFcLong( 0x400b ),          /* 16395 */
/* 144 */ NdrFcShort( 0x2d2 ),          /* Offset= 722 (866) */
/* 146 */ NdrFcLong( 0x400a ),          /* 16394 */
/* 150 */ NdrFcShort( 0x2d0 ),          /* Offset= 720 (870) */
/* 152 */ NdrFcLong( 0x4006 ),          /* 16390 */
/* 156 */ NdrFcShort( 0x2d6 ),          /* Offset= 726 (882) */
/* 158 */ NdrFcLong( 0x4007 ),          /* 16391 */
/* 162 */ NdrFcShort( 0x2cc ),          /* Offset= 716 (878) */
/* 164 */ NdrFcLong( 0x4008 ),          /* 16392 */
/* 168 */ NdrFcShort( 0x2ce ),          /* Offset= 718 (886) */
/* 170 */ NdrFcLong( 0x400d ),          /* 16397 */
/* 174 */ NdrFcShort( 0x2cc ),          /* Offset= 716 (890) */
/* 176 */ NdrFcLong( 0x4009 ),          /* 16393 */
/* 180 */ NdrFcShort( 0x2ca ),          /* Offset= 714 (894) */
/* 182 */ NdrFcLong( 0x6000 ),          /* 24576 */
/* 186 */ NdrFcShort( 0x2c8 ),          /* Offset= 712 (898) */
/* 188 */ NdrFcLong( 0x400c ),          /* 16396 */
/* 192 */ NdrFcShort( 0x2c6 ),          /* Offset= 710 (902) */
/* 194 */ NdrFcLong( 0x10 ),  /* 16 */
/* 198 */ NdrFcShort( 0x8002 ),         /* Simple arm type: FC_CHAR */
/* 200 */ NdrFcLong( 0x12 ),  /* 18 */
/* 204 */ NdrFcShort( 0x8006 ),         /* Simple arm type: FC_SHORT */
/* 206 */ NdrFcLong( 0x13 ),  /* 19 */
/* 210 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 212 */ NdrFcLong( 0x16 ),  /* 22 */
/* 216 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 218 */ NdrFcLong( 0x17 ),  /* 23 */
/* 222 */ NdrFcShort( 0x8008 ),         /* Simple arm type: FC_LONG */
/* 224 */ NdrFcLong( 0xe ),   /* 14 */
/* 228 */ NdrFcShort( 0x2aa ),          /* Offset= 682 (910) */
/* 230 */ NdrFcLong( 0x400e ),          /* 16398 */
/* 234 */ NdrFcShort( 0x2b0 ),          /* Offset= 688 (922) */
/* 236 */ NdrFcLong( 0x4010 ),          /* 16400 */
/* 240 */ NdrFcShort( 0x2ae ),          /* Offset= 686 (926) */
```

```
/* 242 */ NdrFcLong( 0x4012 ),          /* 16402 */
/* 246 */ NdrFcShort( 0x26c ),          /* Offset= 620 (866) */
/* 248 */ NdrFcLong( 0x4013 ),          /* 16403 */
/* 252 */ NdrFcShort( 0x26a ),          /* Offset= 618 (870) */
/* 254 */ NdrFcLong( 0x4016 ),          /* 16406 */
/* 258 */ NdrFcShort( 0x264 ),          /* Offset= 612 (870) */
/* 260 */ NdrFcLong( 0x4017 ),          /* 16407 */
/* 264 */ NdrFcShort( 0x25e ),          /* Offset= 606 (870) */
/* 266 */ NdrFcLong( 0x0 ),   /* 0 */
/* 270 */ NdrFcShort( 0x0 ),  /* Offset= 0 (270) */
/* 272 */ NdrFcLong( 0x1 ),   /* 1 */
/* 276 */ NdrFcShort( 0x0 ),  /* Offset= 0 (276) */
/* 278 */ NdrFcShort( 0xffffffff ),     /* Offset= -1 (277) */
/* 280 */
                                0x15,               /* FC_STRUCT */
                                0x7,                /* 7 */
/* 282 */ NdrFcShort( 0x8 ),  /* 8 */
/* 284 */ 0xb,                 /* FC_HYPER */
                                0x5b,               /* FC_END */
/* 286 */
                                0x12, 0x0,          /* FC_UP */
/* 288 */ NdrFcShort( 0xe ),  /* Offset= 14 (302) */
/* 290 */
                                0x1b,               /* FC_CARRAY */
                                0x1,                /* 1 */
/* 292 */ NdrFcShort( 0x2 ),  /* 2 */
/* 294 */ 0x9,                 /* Corr desc: FC_ULONG */
                                0x0,                /*  */
/* 296 */ NdrFcShort( 0xfffc ),         /* -4 */
/* 298 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 300 */ 0x6,                 /* FC_SHORT */
                                0x5b,               /* FC_END */
/* 302 */
                                0x17,               /* FC_CSTRUCT */
                                0x3,                /* 3 */
/* 304 */ NdrFcShort( 0x8 ),  /* 8 */
/* 306 */ NdrFcShort( 0xfffffff0 ),     /* Offset= -16 (290) */
/* 308 */ 0x8,                 /* FC_LONG */
                                0x8,                /* FC_LONG */
/* 310 */ 0x5c,                /* FC_PAD */
                                0x5b,               /* FC_END */
/* 312 */
                                0x2f,               /* FC_IP */
                                0x5a,               /* FC_CONSTANT_IID */
/* 314 */ NdrFcLong( 0x0 ),   /* 0 */
/* 318 */ NdrFcShort( 0x0 ),  /* 0 */
/* 320 */ NdrFcShort( 0x0 ),  /* 0 */
/* 322 */ 0xc0,                /* 192 */
                                0x0,                /* 0 */
/* 324 */ 0x0,                 /* 0 */
                                0x0,                /* 0 */
/* 326 */ 0x0,                 /* 0 */
                                0x0,                /* 0 */
/* 328 */ 0x0,                 /* 0 */
                                0x46,               /* 70 */
/* 330 */
                                0x2f,               /* FC_IP */
                                0x5a,               /* FC_CONSTANT_IID */
/* 332 */ NdrFcLong( 0x20400 ),         /* 132096 */
/* 336 */ NdrFcShort( 0x0 ),  /* 0 */
/* 338 */ NdrFcShort( 0x0 ),  /* 0 */
/* 340 */ 0xc0,                /* 192 */
                                0x0,                /* 0 */
/* 342 */ 0x0,                 /* 0 */
                                0x0,                /* 0 */
/* 344 */ 0x0,                 /* 0 */
                                0x0,                /* 0 */
/* 346 */ 0x0,                 /* 0 */
                                0x46,               /* 70 */
/* 348 */
                                0x12, 0x10,         /* FC_UP [pointer_deref] */
/* 350 */ NdrFcShort( 0x2 ),  /* Offset= 2 (352) */
/* 352 */
                                0x12, 0x0,          /* FC_UP */
/* 354 */ NdrFcShort( 0x1e6 ),          /* Offset= 486 (840) */
/* 356 */
                                0x2a,               /* FC_ENCAPSULATED_UNION */
                                0x89,               /* 137 */
/* 358 */ NdrFcShort( 0x20 ), /* 32 */
/* 360 */ NdrFcShort( 0xa ),  /* 10 */
/* 362 */ NdrFcLong( 0x8 ),   /* 8 */
/* 366 */ NdrFcShort( 0x50 ), /* Offset= 80 (446) */
/* 368 */ NdrFcLong( 0xd ),   /* 13 */
/* 372 */ NdrFcShort( 0x70 ), /* Offset= 112 (484) */
/* 374 */ NdrFcLong( 0x9 ),   /* 9 */
/* 378 */ NdrFcShort( 0x90 ), /* Offset= 144 (522) */
/* 380 */ NdrFcLong( 0xc ),   /* 12 */
/* 384 */ NdrFcShort( 0xb0 ), /* Offset= 176 (560) */
/* 386 */ NdrFcLong( 0x24 ),  /* 36 */
/* 390 */ NdrFcShort( 0x104 ),          /* Offset= 260 (650) */
/* 392 */ NdrFcLong( 0x800d ),          /* 32781 */
/* 396 */ NdrFcShort( 0x120 ),          /* Offset= 288 (684) */
/* 398 */ NdrFcLong( 0x10 ),  /* 16 */
```

```
/* 402 */ NdrFcShort( 0x13a ),          /* Offset= 314 (716) */
/* 404 */ NdrFcLong( 0x2 ),   /* 2 */
/* 408 */ NdrFcShort( 0x150 ),          /* Offset= 336 (744) */
/* 410 */ NdrFcLong( 0x3 ),   /* 3 */
/* 414 */ NdrFcShort( 0x166 ),          /* Offset= 358 (772) */
/* 416 */ NdrFcLong( 0x14 ),  /* 20 */
/* 420 */ NdrFcShort( 0x17c ),          /* Offset= 380 (800) */
/* 422 */ NdrFcShort( 0xffffffff ),     /* Offset= -1 (421) */
/* 424 */
                           0x21,               /* FC_BOGUS_ARRAY */
                           0x3,                /* 3 */
/* 426 */ NdrFcShort( 0x0 ),  /* 0 */
/* 428 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                           0x0,                /*  */
/* 430 */ NdrFcShort( 0x0 ),  /* 0 */
/* 432 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 434 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 438 */ NdrFcShort( 0x0 ),  /* Corr flags:  */
/* 440 */
                           0x12, 0x0,          /* FC_UP */
/* 442 */ NdrFcShort( 0xffffff74 ),     /* Offset= -140 (302) */
/* 444 */ 0x5c,               /* FC_PAD */
                           0x5b,               /* FC_END */
/* 446 */
                           0x1a,               /* FC_BOGUS_STRUCT */
                           0x3,                /* 3 */
/* 448 */ NdrFcShort( 0x10 ), /* 16 */
/* 450 */ NdrFcShort( 0x0 ),  /* 0 */
/* 452 */ NdrFcShort( 0x6 ),  /* Offset= 6 (458) */
/* 454 */ 0x8,                /* FC_LONG */
                           0x39,               /* FC_ALIGNM8 */
/* 456 */ 0x36,               /* FC_POINTER */
                           0x5b,               /* FC_END */
/* 458 */
                           0x11, 0x0,          /* FC_RP */
/* 460 */ NdrFcShort( 0xffffffdc ),     /* Offset= -36 (424) */
/* 462 */
                           0x21,               /* FC_BOGUS_ARRAY */
                           0x3,                /* 3 */
/* 464 */ NdrFcShort( 0x0 ),  /* 0 */
/* 466 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                           0x0,                /*  */
/* 468 */ NdrFcShort( 0x0 ),  /* 0 */
/* 470 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 472 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 476 */ NdrFcShort( 0x0 ),  /* Corr flags:  */
/* 478 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                           0x0,                /* 0 */
/* 480 */ NdrFcShort( 0xffffff58 ),     /* Offset= -168 (312) */
/* 482 */ 0x5c,               /* FC_PAD */
                           0x5b,               /* FC_END */
/* 484 */
                           0x1a,               /* FC_BOGUS_STRUCT */
                           0x3,                /* 3 */
/* 486 */ NdrFcShort( 0x10 ), /* 16 */
/* 488 */ NdrFcShort( 0x0 ),  /* 0 */
/* 490 */ NdrFcShort( 0x6 ),  /* Offset= 6 (496) */
/* 492 */ 0x8,                /* FC_LONG */
                           0x39,               /* FC_ALIGNM8 */
/* 494 */ 0x36,               /* FC_POINTER */
                           0x5b,               /* FC_END */
/* 496 */
                           0x11, 0x0,          /* FC_RP */
/* 498 */ NdrFcShort( 0xffffffdc ),     /* Offset= -36 (462) */
/* 500 */
                           0x21,               /* FC_BOGUS_ARRAY */
                           0x3,                /* 3 */
/* 502 */ NdrFcShort( 0x0 ),  /* 0 */
/* 504 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                           0x0,                /*  */
/* 506 */ NdrFcShort( 0x0 ),  /* 0 */
/* 508 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 510 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 514 */ NdrFcShort( 0x0 ),  /* Corr flags:  */
/* 516 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                           0x0,                /* 0 */
/* 518 */ NdrFcShort( 0xffffff44 ),     /* Offset= -188 (330) */
/* 520 */ 0x5c,               /* FC_PAD */
                           0x5b,               /* FC_END */
/* 522 */
                           0x1a,               /* FC_BOGUS_STRUCT */
                           0x3,                /* 3 */
/* 524 */ NdrFcShort( 0x10 ), /* 16 */
/* 526 */ NdrFcShort( 0x0 ),  /* 0 */
/* 528 */ NdrFcShort( 0x6 ),  /* Offset= 6 (534) */
/* 530 */ 0x8,                /* FC_LONG */
                           0x39,               /* FC_ALIGNM8 */
/* 532 */ 0x36,               /* FC_POINTER */
                           0x5b,               /* FC_END */
/* 534 */
                           0x11, 0x0,          /* FC_RP */
/* 536 */ NdrFcShort( 0xffffffdc ),     /* Offset= -36 (500) */
/* 538 */
```

```
                            0x21,               /* FC_BOGUS_ARRAY */
                            0x3,                /* 3 */
/* 540 */ NdrFcShort( 0x0 ),  /* 0 */
/* 542 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 544 */ NdrFcShort( 0x0 ),  /* 0 */
/* 546 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 548 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 552 */ NdrFcShort( 0x0 ),  /* Corr flags:  */
/* 554 */
                            0x12, 0x0,          /* FC_UP */
/* 556 */ NdrFcShort( 0x176 ),          /* Offset= 374 (930) */
/* 558 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 560 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 562 */ NdrFcShort( 0x10 ), /* 16 */
/* 564 */ NdrFcShort( 0x0 ),  /* 0 */
/* 566 */ NdrFcShort( 0x6 ),  /* Offset= 6 (572) */
/* 568 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 570 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 572 */
                            0x11, 0x0,          /* FC_RP */
/* 574 */ NdrFcShort( 0xfffffffc ),     /* Offset= -36 (538) */
/* 576 */
                            0x2f,               /* FC_IP */
                            0x5a,               /* FC_CONSTANT_IID */
/* 578 */ NdrFcLong( 0x2f ),  /* 47 */
/* 582 */ NdrFcShort( 0x0 ),  /* 0 */
/* 584 */ NdrFcShort( 0x0 ),  /* 0 */
/* 586 */ 0xc0,               /* 192 */
                            0x0,                /* 0 */
/* 588 */ 0x0,                /* 0 */
                            0x0,                /* 0 */
/* 590 */ 0x0,                /* 0 */
                            0x0,                /* 0 */
/* 592 */ 0x0,                /* 0 */
                            0x46,               /* 70 */
/* 594 */
                            0x1b,               /* FC_CARRAY */
                            0x0,                /* 0 */
/* 596 */ NdrFcShort( 0x1 ),  /* 1 */
/* 598 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 600 */ NdrFcShort( 0x4 ),  /* 4 */
/* 602 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 604 */ 0x1,                /* FC_BYTE */
                            0x5b,               /* FC_END */
/* 606 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 608 */ NdrFcShort( 0x18 ), /* 24 */
/* 610 */ NdrFcShort( 0x0 ),  /* 0 */
/* 612 */ NdrFcShort( 0xc ),  /* Offset= 12 (624) */
/* 614 */ 0x8,                /* FC_LONG */
                            0x8,                /* FC_LONG */
/* 616 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                            0x0,                /* 0 */
/* 618 */ NdrFcShort( 0xffffffd6 ),     /* Offset= -42 (576) */
/* 620 */ 0x39,               /* FC_ALIGNM8 */
                            0x36,               /* FC_POINTER */
/* 622 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 624 */
                            0x12, 0x0,          /* FC_UP */
/* 626 */ NdrFcShort( 0xffffffe0 ),     /* Offset= -32 (594) */
/* 628 */
                            0x21,               /* FC_BOGUS_ARRAY */
                            0x3,                /* 3 */
/* 630 */ NdrFcShort( 0x0 ),  /* 0 */
/* 632 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 634 */ NdrFcShort( 0x0 ),  /* 0 */
/* 636 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 638 */ NdrFcLong( 0xffffffff ),      /* -1 */
/* 642 */ NdrFcShort( 0x0 ),  /* Corr flags:  */
/* 644 */
                            0x12, 0x0,          /* FC_UP */
/* 646 */ NdrFcShort( 0xffffffd8 ),     /* Offset= -40 (606) */
/* 648 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 650 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 652 */ NdrFcShort( 0x10 ), /* 16 */
/* 654 */ NdrFcShort( 0x0 ),  /* 0 */
/* 656 */ NdrFcShort( 0x6 ),  /* Offset= 6 (662) */
/* 658 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 660 */ 0x36,               /* FC_POINTER */
```

```
                            0x5b,               /* FC_END */
/* 662 */
                            0x11, 0x0,          /* FC_RP */
/* 664 */ NdrFcShort( 0xffffffdc ),     /* Offset= -36 (628) */
/* 666 */
                            0x1d,               /* FC_SMFARRAY */
                            0x0,                /* 0 */
/* 668 */ NdrFcShort( 0x8 ),  /* 8 */
/* 670 */ 0x2,                /* FC_CHAR */
                            0x5b,               /* FC_END */
/* 672 */
                            0x15,               /* FC_STRUCT */
                            0x3,                /* 3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ 0x8,                /* FC_LONG */
                            0x6,                /* FC_SHORT */
/* 678 */ 0x6,                /* FC_SHORT */
                            0x4c,               /* FC_EMBEDDED_COMPLEX */
/* 680 */ 0x0,                /* 0 */
                            NdrFcShort( 0xfffffff1 ),     /* Offset= -15 (666) */
                            0x5b,               /* FC_END */
/* 684 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 686 */ NdrFcShort( 0x20 ), /* 32 */
/* 688 */ NdrFcShort( 0x0 ),  /* 0 */
/* 690 */ NdrFcShort( 0xa ),  /* Offset= 10 (700) */
/* 692 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 694 */ 0x36,               /* FC_POINTER */
                            0x4c,               /* FC_EMBEDDED_COMPLEX */
/* 696 */ 0x0,                /* 0 */
                            NdrFcShort( 0xffffffe7 ),     /* Offset= -25 (672) */
                            0x5b,               /* FC_END */
/* 700 */
                            0x11, 0x0,          /* FC_RP */
/* 702 */ NdrFcShort( 0xffffff10 ),     /* Offset= -240 (462) */
/* 704 */
                            0x1b,               /* FC_CARRAY */
                            0x0,                /* 0 */
/* 706 */ NdrFcShort( 0x1 ),  /* 1 */
/* 708 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 710 */ NdrFcShort( 0x0 ),  /* 0 */
/* 712 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 714 */ 0x1,                /* FC_BYTE */
                            0x5b,               /* FC_END */
/* 716 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 718 */ NdrFcShort( 0x10 ), /* 16 */
/* 720 */ NdrFcShort( 0x0 ),  /* 0 */
/* 722 */ NdrFcShort( 0x6 ),  /* Offset= 6 (728) */
/* 724 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 726 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 728 */
                            0x12, 0x0,          /* FC_UP */
/* 730 */ NdrFcShort( 0xfffffffe6 ),     /* Offset= -26 (704) */
/* 732 */
                            0x1b,               /* FC_CARRAY */
                            0x1,                /* 1 */
/* 734 */ NdrFcShort( 0x2 ),  /* 2 */
/* 736 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 738 */ NdrFcShort( 0x0 ),  /* 0 */
/* 740 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 742 */ 0x6,                /* FC_SHORT */
                            0x5b,               /* FC_END */
/* 744 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 746 */ NdrFcShort( 0x10 ), /* 16 */
/* 748 */ NdrFcShort( 0x0 ),  /* 0 */
/* 750 */ NdrFcShort( 0x6 ),  /* Offset= 6 (756) */
/* 752 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 754 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 756 */
                            0x12, 0x0,          /* FC_UP */
/* 758 */ NdrFcShort( 0xfffffffe6 ),     /* Offset= -26 (732) */
/* 760 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 762 */ NdrFcShort( 0x4 ),  /* 4 */
/* 764 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 766 */ NdrFcShort( 0x0 ),  /* 0 */
/* 768 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 770 */ 0x8,                /* FC_LONG */
                            0x5b,               /* FC_END */
```

```
/* 772 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 774 */ NdrFcShort( 0x10 ), /* 16 */
/* 776 */ NdrFcShort( 0x0 ),  /* 0 */
/* 778 */ NdrFcShort( 0x6 ),  /* Offset= 6 (784) */
/* 780 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 782 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 784 */
                            0x12, 0x0,          /* FC_UP */
/* 786 */ NdrFcShort( 0xfffffff6 ),     /* Offset= -26 (760) */
/* 788 */
                            0x1b,               /* FC_CARRAY */
                            0x7,                /* 7 */
/* 790 */ NdrFcShort( 0x8 ),  /* 8 */
/* 792 */ 0x19,               /* Corr desc:  field pointer, FC_ULONG */
                            0x0,                /*  */
/* 794 */ NdrFcShort( 0x0 ),  /* 0 */
/* 796 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 798 */ 0xb,                /* FC_HYPER */
                            0x5b,               /* FC_END */
/* 800 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 802 */ NdrFcShort( 0x10 ), /* 16 */
/* 804 */ NdrFcShort( 0x0 ),  /* 0 */
/* 806 */ NdrFcShort( 0x6 ),  /* Offset= 6 (812) */
/* 808 */ 0x8,                /* FC_LONG */
                            0x39,               /* FC_ALIGNM8 */
/* 810 */ 0x36,               /* FC_POINTER */
                            0x5b,               /* FC_END */
/* 812 */
                            0x12, 0x0,          /* FC_UP */
/* 814 */ NdrFcShort( 0xfffffff6 ),     /* Offset= -26 (788) */
/* 816 */
                            0x15,               /* FC_STRUCT */
                            0x3,                /* 3 */
/* 818 */ NdrFcShort( 0x8 ),  /* 8 */
/* 820 */ 0x8,                /* FC_LONG */
                            0x8,                /* FC_LONG */
/* 822 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 824 */
                            0x1b,               /* FC_CARRAY */
                            0x3,                /* 3 */
/* 826 */ NdrFcShort( 0x8 ),  /* 8 */
/* 828 */ 0x7,                /* Corr desc: FC_USHORT */
                            0x0,                /*  */
/* 830 */ NdrFcShort( 0xffc8 ),         /* -56 */
/* 832 */ NdrFcShort( 0x1 ),  /* Corr flags:  early, */
/* 834 */ 0x4c,               /* FC_EMBEDDED_COMPLEX */
                            0x0,                /* 0 */
/* 836 */ NdrFcShort( 0xffffffec ),     /* Offset= -20 (816) */
/* 838 */ 0x5c,               /* FC_PAD */
                            0x5b,               /* FC_END */
/* 840 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 842 */ NdrFcShort( 0x38 ), /* 56 */
/* 844 */ NdrFcShort( 0xffffffec ),     /* Offset= -20 (824) */
/* 846 */ NdrFcShort( 0x0 ),  /* Offset= 0 (846) */
/* 848 */ 0x6,                /* FC_SHORT */
                            0x6,                /* FC_SHORT */
/* 850 */ 0x38,               /* FC_ALIGNM4 */
                            0x8,                /* FC_LONG */
/* 852 */ 0x8,                /* FC_LONG */
                            0x4c,               /* FC_EMBEDDED_COMPLEX */
/* 854 */ 0x4,                /* 4 */
                            NdrFcShort( 0xfffffe0d ),     /* Offset= -499 (356) */
                            0x5b,               /* FC_END */
/* 858 */
                            0x12, 0x0,          /* FC_UP */
/* 860 */ NdrFcShort( 0xfffffff02 ),     /* Offset= -254 (606) */
/* 862 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 864 */ 0x1,                /* FC_BYTE */
                            0x5c,               /* FC_PAD */
/* 866 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 868 */ 0x6,                /* FC_SHORT */
                            0x5c,               /* FC_PAD */
/* 870 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 872 */ 0x8,                /* FC_LONG */
                            0x5c,               /* FC_PAD */
/* 874 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
/* 876 */ 0xa,                /* FC_FLOAT */
                            0x5c,               /* FC_PAD */
/* 878 */
                            0x12, 0x8,          /* FC_UP [simple_pointer] */
```

```
/* 880 */ 0xc,              /* FC_DOUBLE */
                            0x5c,              /* FC_PAD */
/* 882 */
                            0x12, 0x0,         /* FC_UP */
/* 884 */ NdrFcShort( 0xffffffda4 ),     /* Offset= -604 (280) */
/* 886 */
                            0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xffffffda6 ),     /* Offset= -602 (286) */
/* 890 */
                            0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xffffffdbc ),     /* Offset= -580 (312) */
/* 894 */
                            0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xffffffdca ),     /* Offset= -566 (330) */
/* 898 */
                            0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xffffffdd8 ),     /* Offset= -552 (348) */
/* 902 */
                            0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 904 */ NdrFcShort( 0x2 ),  /* Offset= 2 (906) */
/* 906 */
                            0x12, 0x0,         /* FC_UP */
/* 908 */ NdrFcShort( 0x16 ), /* Offset= 22 (930) */
/* 910 */
                            0x15,              /* FC_STRUCT */
                            0x7,               /* 7 */
/* 912 */ NdrFcShort( 0x10 ), /* 16 */
/* 914 */ 0x6,              /* FC_SHORT */
                            0x1,               /* FC_BYTE */
/* 916 */ 0x1,              /* FC_BYTE */
                            0x38,              /* FC_ALIGNM4 */
/* 918 */ 0x8,              /* FC_LONG */
                            0x39,              /* FC_ALIGNM8 */
/* 920 */ 0xb,              /* FC_HYPER */
                            0x5b,              /* FC_END */
/* 922 */
                            0x12, 0x0,         /* FC_UP */
/* 924 */ NdrFcShort( 0xfffffff2 ),     /* Offset= -14 (910) */
/* 926 */
                            0x12, 0x8,         /* FC_UP [simple_pointer] */
/* 928 */ 0x2,              /* FC_CHAR */
                            0x5c,              /* FC_PAD */
/* 930 */
                            0x1a,              /* FC_BOGUS_STRUCT */
                            0x7,               /* 7 */
/* 932 */ NdrFcShort( 0x20 ), /* 32 */
/* 934 */ NdrFcShort( 0x0 ),  /* 0 */
/* 936 */ NdrFcShort( 0x0 ),  /* Offset= 0 (936) */
/* 938 */ 0x8,              /* FC_LONG */
                            0x8,               /* FC_LONG */
/* 940 */ 0x6,              /* FC_SHORT */
                            0x6,               /* FC_SHORT */
/* 942 */ 0x6,              /* FC_SHORT */
                            0x6,               /* FC_SHORT */
/* 944 */ 0x4c,             /* FC_EMBEDDED_COMPLEX */
                            0x0,               /* 0 */
/* 946 */ NdrFcShort( 0xfffffc54 ),     /* Offset= -940 (6) */
/* 948 */ 0x5c,             /* FC_PAD */
                            0x5b,              /* FC_END */
/* 950 */ 0xb4,             /* FC_USER_MARSHAL */
                            0x83,              /* 131 */
/* 952 */ NdrFcShort( 0x0 ),  /* 0 */
/* 954 */ NdrFcShort( 0x18 ), /* 24 */
/* 956 */ NdrFcShort( 0x0 ),  /* 0 */
/* 958 */ NdrFcShort( 0xfffffc44 ),     /* Offset= -956 (2) */
/* 960 */
                            0x11, 0x4,         /* FC_RP [alloced_on_stack] */
/* 962 */ NdrFcShort( 0x6 ),  /* Offset= 6 (968) */
/* 964 */
                            0x13, 0x0,         /* FC_OP */
/* 966 */ NdrFcShort( 0xfffffffdc ),     /* Offset= -36 (930) */
/* 968 */ 0xb4,             /* FC_USER_MARSHAL */
                            0x83,              /* 131 */
/* 970 */ NdrFcShort( 0x0 ),  /* 0 */
/* 972 */ NdrFcShort( 0x18 ), /* 24 */
/* 974 */ NdrFcShort( 0x0 ),  /* 0 */
/* 976 */ NdrFcShort( 0xfffffff4 ),     /* Offset= -12 (964) */

                            0x0
        }
    };

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

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

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


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

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{

    if(!_tpcc_com_ps_CHECK_IID(0))
        {
        *pIndex = 0;
        return 1;
        }

    return 0;
}

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


#endif /* defined(_M_IA64) || defined(_M_AXP64)*/
```

common/txnlog/include/rtetime.h

```
/* FILE: rtetime.h : header file
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Charles Levine, Philip Durr
 *                           Microsoft Corp.
 */

#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         ConvertTo(x64)Time(int iYear, int iMonth, int iDay, int iHour, int iMinute, int iSecond);
JULIAN_TIME         Get(x64)Time(LPSYSTEMTIME lpInitTime);
int                       JulianDay( int yr, int mm, int dd );
void                JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd, int *hh, int *mi, int *ss );
void                JulianToCalendar( int day, int* yr, int* mm, int* dd );
}
```

common/txnlog/include/spinlock.h

```
/*        FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 *                           Microsoft Corp.
 */

#ifndef _INC_Spinlock
```

```
const LONG LockClosed         = 1;
const LONG LockOpen           = 0;

/************************************************************
 *
 *   Spinlock and Semaphore locking.
 *
 *   This class provides a very conservative locking scheme.
 *   The assumption behind the code is that locks will be
 *   held for a very short time.  When a lock is taken a memory
 *   location is exchanged.  All other threads that want this
 *   lock wait by spinning and sometimes sleeping on a semaphore
 *   until it becomes free again.  The only other choice is not
 *   to wait at all and move on to do something else.  This
 *   module should normally be used in conjunction with cache
 *   aligned memory in minimize cache line misses.
 *
 ************************************************************/

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

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

          public:
                    // Public functions.

                    Spinlock( void );

                    inline BOOL ClaimLock( BOOL Wait = TRUE );
                    inline void ReleaseLock( void );
                    ~Spinlock( void );
                    // Disabled operations.
                    Spinlock( const Spinlock & Copy );
                    void operator=( const Spinlock & Copy );

          private:
                    // Private functions.
                    inline BOOL ClaimSpinlock( volatile LONG *sl );
                    void WaitForLock( void );
                    void WakeAllSleepers( void );
};

/************************************************************
 *
 *   A guaranteed atomic exchange.
 *
 *   An attempt is made to claim the Spinlock.  This action is
 *   guaranteed to be atomic.
 *
 ************************************************************/

inline BOOL Spinlock::ClaimSpinlock( volatile LONG *Spinlock )
{
          #ifdef _DEBUG
                    InterlockedIncrement( (LPLONG) & TotalLocks );
          #endif
          return ( ((*Spinlock) == LockOpen) && (InterlockedExchange( (LPLONG)Spinlock, LockClosed ) ==
LockOpen) );
}

/************************************************
 *
 *   Claim the Spinlock.
 *
 *   Claim the lock if available else wait or exit.
 *
 ************************************************/

inline BOOL Spinlock::ClaimLock( BOOL Wait )
{
          if ( ! ClaimSpinlock( (volatile LONG*) & m_Spinlock ) )
          {
                    if ( Wait )
                              WaitForLock();
                    return Wait;
          }
          return TRUE;
}

/******************************************************
 *
 *   Release the Spinlock.
```

```
           *
           *   Release the lock and if needed wakeup any sleepers.
           *
           ******************************************************/

          inline void Spinlock::ReleaseLock( void )
          {
                    m_Spinlock = LockOpen;
                    if ( Waiting > 0 )
                              WakeAllSleepers();
          }

          #define _INC_Spinlock

#endif
```

common/txnlog/include/txnlog.h

```
/*        FILE:               TXNLOG.H
 *                                    Microsoft TPC-C Kit Ver. 4.10.000
 *
 *                                    NOTE: this file is RTE specific and should not be included
 *                                    in Full Disclosure Reports.
 *
 *                                    Copyright Microsoft, 1999
 *
 *        PURPOSE:  Structure definitions for logging delivery txn completion stats.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 */


    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

          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
                WORD      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;         //
        } 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)
                WORD      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
        } TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;



#define             TXN_LOG_VERSION             1
#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
```

```
                    // the record map provides a fast way to get close to a particular timestamp in a sorted log file.
//                  struct
//                  {
//                            JULIAN_TIME                   TS;                           // timestamp of record
//                            int                                   iPos;                 // byte position in file
//                  }                                     RecMap[RecMapSize];
//#define           RecMapSize                            200

          } TXN_LOG_HEADER, *PTXN_LOG_HEADER;



#define   READ_BUFFER_SIZE              64*1024
#define   WRITE_BUFFER_SIZE              8*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_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.
                    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;
                    int                                   iSavePtNextRec;

                    JULIAN_TIME         lastTS;                                     //when writing sorted output,
used to verify records are sorted
                    BOOL                bWrite;                                     //writing log file

                    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

                    BYTE                *pCurrent;                                  //ptr to current buffer
                    BYTE                *pBuffer[MAX_NUM_BUFFERS];

                    PTXN_RECORD_HEADER  *TxnArray;                    //transaction record pointer array for sort

                    DWORD               dwError;
                    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

                    Spinlock  Spin;                                           //spin lock to protect the txn log file
buffers

                    int Write(BYTE *ptr, DWORD Size);
                    static void LogFileIO(CTxnLog *);

          public:
                    CTxnLog::CTxnLog(LPCTSTR szFileName, DWORD dwOpts);
                    ~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 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(int index);

                    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 CTPCC_DBLIB_ERRS
                    {
                              ERR_BAD_FILE_FORMAT = 1,      // "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 ) { m_errno = iErr; };

                    int                m_errno;

                    int ErrorType() {return ERR_TYPE_TXNLOG;};
                    int ErrorNum() {return m_errno;};

                    // TODO: need to complete...
                    char *ErrorText() {return "";};
};
```

Appendix B - Database Design

Build Scripts

```
ECHO OFF
::-----------------------------------------------------------------------------------------
::---  FILE:       SETUP.CMD
::---              Microsoft TPC-C Kit Ver. 4.62
::---              Copyright Microsoft, 2001, 2002, 2005
::---              All Rights Reserved
::---
::---  PURPOSE:    Calls RunSQLCfg.sql to configure SQL Server
::---
::---  ARGUMENTS:  /? displays help for SETUP
::---
::-----------------------------------------------------------------------------------------
@cscript SetupScripts\setup.vbs //H:CScript //I %1 %2 %3 %4 %5 %6 %7
-------------------------------------------------------------------
--                                                               --
--  File:   TABLES.SQL                                           --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62              --
--          Copyright Microsoft, 2005                            --
--                                                               --
--          Creates TPC-C tables                                 --
-------------------------------------------------------------------

SET ANSI_NULL_DFLT_OFF ON
GO

USE tpcc
GO

-----------------------------------
-- Remove all existing TPC-C tables
-----------------------------------
if exists ( select name from sysobjects where name = 'warehouse' )
          drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
          drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
          drop table customer
```

```
go
if exists ( select name from sysobjects where name = 'history' )
         drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
         drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
         drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
         drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
         drop table item
go
if exists ( select name from sysobjects where name = 'stock' )
         drop table stock
go

--------------------
-- Create new tables
--------------------
create table warehouse
(
          w_id               int,
          w_ytd              money,
          w_tax              smallmoney,
          w_name             char(10),
          w_street_1         char(20),
          w_street_2         char(20),
          w_city             char(20),
          w_state            char(2),
          w_zip              char(9)
) on 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
-----------------------------------------------------------------------
--                                                                   --
--  File:   IDXCUSCL.SQL                                             --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                  --
--          Copyright Microsoft, 2005                                --
--                                                                   --
--          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
-----------------------------------------------------------------
--                                                             --
--  File:   IDXCUSNC.SQL                                       --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62            --
--          Copyright Microsoft, 2005                          --
--                                                             --
--          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
-----------------------------------------------------------------
--                                                             --
--  File:   IDXDISCL.SQL                                       --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62            --
--          Copyright Microsoft, 2005                          --
--                                                             --
--          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:   IDXHISCL.SQL                                       --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62            --
--          Copyright Microsoft, 2005                          --
--                                                             --
--          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
----------------------------------------------------------------------
--                                                                  --
--  File:   IDXITMCL.SQL                                            --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                 --
--          Copyright Microsoft, 2005                               --
--                                                                  --
--          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
----------------------------------------------------------------------
--                                                                  --
--  File:   IDXNODCL.SQL                                            --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                 --
--          Copyright Microsoft, 2005                               --
--                                                                  --
--          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
----------------------------------------------------------------------
--                                                                  --
--  File:   IDXODLCL.SQL                                            --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62                 --
--          Copyright Microsoft, 2005                               --
--                                                                  --
--          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
------------------------------------------------------------------
--                                                              --
--  File:   IDXODLCL.SQL                                        --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62             --
--          Copyright Microsoft, 2005                           --
--                                                              --
--          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
------------------------------------------------------------------
--                                                              --
--  File:   IDXORDNC.SQL                                        --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62             --
--          Copyright Microsoft, 2005                           --
--                                                              --
--          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
------------------------------------------------------------------
--                                                              --
--  File:   IDXSTKCL.SQL                                        --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62             --
--          Copyright Microsoft, 2005                           --
--                                                              --
--          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
-----------------------------------------------------------------
--                                                             --
--  File:   IDXWARCL.SQL                                       --
--          Microsoft TPC-C Benchmark Kit Ver. 4.62            --
--          Copyright Microsoft, 2005                          --
--                                                             --
--          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


dbopt1.sql

--  File:     DBOPT1.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Sets database options for data load


use master
go

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

use tpcc
go

checkpoint
go







dbopt2.sql

--  File:     DBOPT2.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.00
--            Copyright Microsoft, 1996
--  Purpose:  Resets database options after data load


use master
go

sp_dboption tpcc,'select ',false
go

sp_dboption tpcc,'trunc. ',false
go

use tpcc
go

checkpoint
go

sp_configure allow,1
```

```
go

reconfigure with override
go

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

sp_indexoption 'customer','AllowPageLocks',FALSE
go
sp_indexoption 'district','AllowPageLocks',FALSE
go
sp_indexoption 'warehouse','AllowPageLocks',FALSE
go
sp_indexoption 'stock','AllowPageLocks',FALSE
go
sp_indexoption 'order_line','AllowPageLocks',FALSE
go
sp_indexoption 'orders','AllowPageLocks',FALSE
go
sp_indexoption 'new_order','AllowRowLocks',FALSE
go
sp_indexoption 'item','AllowRowLocks',FALSE
go
sp_indexoption 'item','AllowPageLocks',FALSE
go

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '   Lockflag = 0 ==> No pre-pecified 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,0
go

reconfigure with override
go

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

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


dbopt3.sql

use tpcc
go
sp_indexoption 'orders','AllowPagelocks',TRUE
go
sp_indexoption 'orders','AllowRowlocks',FALSE
go
sp_indexoption 'order_line','AllowPagelocks',TRUE
go
sp_indexoption 'order_line','AllowRowlocks',FALSE
go
-----------------------------------------------------------------------
-----------------------------------------------------------------------
--                                                                   --
--  File:  BACKUP.SQL                                                --
--         Microsoft TPC-C Benchmark Kit Ver. 4.51                   --
--         Copyright Microsoft, 2003                                 --
--                                                                   --
-----------------------------------------------------------------------

declare   @startdate          datetime,
          @enddate  datetime
```

```
select    @startdate = getdate()
select    'Start date:',
                  convert(varchar(30),@startdate, 21)

dump database tpcc to tpccback1, tpccback2 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
-----------------------------------------------------------------
--                                                             --
--  File:  RESTORE.SQL                                         --
--         Microsoft TPC-C Benchmark Kit Ver. 4.51             --
--         Copyright Microsoft, 2003                           --
--                                                             --
-----------------------------------------------------------------

declare   @startdate          datetime,
          @enddate  datetime

select    @startdate = getdate()
select    'Start date:',
                  convert(varchar(30),@startdate, 21)

load database tpcc from tpccback1, tpccback2 WITH REPLACE

select    @enddate = getdate()
select    'End date: ',
                  convert(varchar(30),@enddate, 21)
select    'Elapsed time (in seconds): ',
                  datediff(second, @startdate, @enddate)

go
-----------------------------------------------------------------
--                                                             --
--        File:     CREATEDB.SQL                               --
--                  Microsoft TPC-C Benchmark Kit Ver. 4.50          --
--                  Copyright Microsoft, 2003                        --
--                                                             --
--                  Creates 5512 warehouse database            --
--                                                             --
-----------------------------------------------------------------

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))

insert    into tpcc_timer values (0,0)
go

----------------------
-- Store starting time
----------------------
update    tpcc_timer
set       start_date          = (select convert(char(30), getdate(), 21))
go

----------------------------
-- create main database files
----------------------------
CREATE DATABASE tpcc
ON PRIMARY
(         NAME                = MSSQL_tpcc_root,
          FILENAME  = 'C:\MSSQL_tpcc_root.mdf',
          SIZE                = 8MB,
          FILEGROWTH          = 0),
FILEGROUP MSSQL_misc_fg
(         NAME                = MSSQL_misc1,
          FILENAME  = 'O:',
          SIZE                = 49731MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_misc2,
          FILENAME  = 'E:',
          SIZE                = 60388MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_misc3,
          FILENAME  = 'L:',
```

```
          SIZE                = 49731MB,
          FILEGROWTH          = 0),
FILEGROUP MSSQL_cs_fg
(         NAME                = MSSQL_cs1,
          FILENAME  = 'P:',
          SIZE                = 101176MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_cs2,
          FILENAME  = 'W:',
          SIZE                = 122857MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL_cs3,
          FILENAME  = 'V:',
          SIZE                = 101176MB,
          FILEGROWTH          = 0)
LOG ON
(         NAME                = MSSQL_tpcc_log,
          FILENAME  = 'B:',
          SIZE                = 260000MB,
          FILEGROWTH          = 0)
COLLATE Latin1_General_BIN
go

--------------------
-- Store ending time
--------------------
update    tpcc_timer
set       end_date  = (select convert(char(30), getdate(), 21))
go

select 'Elapsed time (in seconds): ', datediff(second,(select start_date from tpcc_timer),(select end_date from
tpcc_timer))

-------------------------
-- remove temporary table
-------------------------
if exists ( select name from sysobjects where name = 'tpcc_timer' )
          drop table tpcc_timer
go
-----------------------------------------------------------------
--                                                             --
--  File:  BACKUPDEV.SQL                                       --
--         Microsoft TPC-C Benchmark Kit Ver. 4.51             --
--         Copyright Microsoft, 2003                           --
--                                                             --
-----------------------------------------------------------------

use master
go

------------------------
-- create backup devices
------------------------
exec sp_addumpdevice 'disk','tpccback1','T:\tpccback1.bak'
go
exec sp_addumpdevice 'disk','tpccback2','R:\tpccback2.bak'
go
-----------------------------------------------------------------
--                                                             --
--  File:  REMOVEDB.SQL                                        --
--         Microsoft TPC-C Benchmark Kit Ver. 4.51             --
--         Copyright Microsoft, 2003                           --
--                                                             --
-----------------------------------------------------------------

use master
go

-----------------------------------------------
-- remove any existing database and backup files
-----------------------------------------------
exec sp_dbremove tpcc, dropdev
go

exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
go
```

Stored Procedures

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 availible order id and update
--  plus initialize local variables
-------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
        SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
```

```
        SELECT  @li_id = CASE @li_no
                                WHEN 1 THEN @i_id1
                                WHEN 2 THEN @i_id2
                                WHEN 3 THEN @i_id3
                                WHEN 4 THEN @i_id4
                                WHEN 5 THEN @i_id5
                                WHEN 6 THEN @i_id6
                                WHEN 7 THEN @i_id7
                                WHEN 8 THEN @i_id8
                                WHEN 9 THEN @i_id9
                                WHEN 10 THEN @i_id10
                                WHEN 11 THEN @i_id11
                                WHEN 12 THEN @i_id12
                                WHEN 13 THEN @i_id13
                                WHEN 14 THEN @i_id14
                                WHEN 15 THEN @i_id15
                        END,

                @li_s_w_id = CASE @li_no
                                WHEN 1 THEN @s_w_id1
                                WHEN 2 THEN @s_w_id2
                                WHEN 3 THEN @s_w_id3
                                WHEN 4 THEN @s_w_id4
                                WHEN 5 THEN @s_w_id5
                                WHEN 6 THEN @s_w_id6
                                WHEN 7 THEN @s_w_id7
                                WHEN 8 THEN @s_w_id8
                                WHEN 9 THEN @s_w_id9
                                WHEN 10 THEN @s_w_id10
                                WHEN 11 THEN @s_w_id11
                                WHEN 12 THEN @s_w_id12
                                WHEN 13 THEN @s_w_id13
                                WHEN 14 THEN @s_w_id14
                                WHEN 15 THEN @s_w_id15
                            END,

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

--------------------------------------
--  get item data (no one updates item)
--------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

-----------------------
--  update stock values
-----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                    CASE WHEN (s_quantity - @li_qty < 10) THEN 91 ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                    CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1 END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                    WHEN 1  THEN s_dist_01
                                    WHEN 2  THEN s_dist_02
                                    WHEN 3  THEN s_dist_03
                                    WHEN 4  THEN s_dist_04
                                    WHEN 5  THEN s_dist_05
                                    WHEN 6  THEN s_dist_06
                                    WHEN 7  THEN s_dist_07
                                    WHEN 8  THEN s_dist_08
                                    WHEN 9  THEN s_dist_09
                                    WHEN 10 THEN s_dist_10
                                  END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------
        IF (@@rowcount > 0)
```

```
        BEGIN
------------------------------------------------------------
--  insert order_line data (using data from item and stock)
------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)

---------------------------------
--  send line-item data to client
---------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) > 0) AND
                                                                        (patindex('%ORIGINAL%',@s_data) > 0) )
                                                                THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                    END
                    ELSE
                    BEGIN
------------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
------------------------------------------------------------
            SELECT  '',0,'',0,0
            SELECT  @commit_flag    = 0
                    END
          END
-------------------------------------------------------
--  get customer last name, discount, and credit rating
-------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

-------------------------------------
--  insert fresh row into orders table
-------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                    COMMIT TRANSACTION n
    ELSE
-------------------------------
--  all that work for nuthin!!!
-------------------------------
        ROLLBACK TRANSACTION n

-------------------------------
--  return order data to client
-------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
```

```
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
------------------------------------------------------------------
--                                                              --
--  File:   PAYMENT.SQL                                         --
--          Microsoft TPC-C Benchmark Kit Ver. 4.68             --
--          Copyright Microsoft, 2006                           --
--                                                              --
--          Creates payment stored procedure                    --
--                                                              --
--          Interface Level:    4.20.000                        --
--                                                              --
------------------------------------------------------------------
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_payment' )
    DROP PROCEDURE tpcc_payment
GO

CREATE PROCEDURE    tpcc_payment
                        @w_id       int,
                        @c_w_id     int,
                        @h_amount   smallmoney,
                        @d_id       tinyint,
                        @c_d_id     tinyint,
                        @c_id       int,
                        @c_last     char(16) = ""

AS
DECLARE @w_street_1     char(20),
        @w_street_2     char(20),
        @w_city         char(20),
        @w_state        char(2),
        @w_zip          char(9),
        @w_name         char(10),
        @d_street_1     char(20),
        @d_street_2     char(20),
        @d_city         char(20),
        @d_state        char(2),
        @d_zip          char(9),
        @d_name         char(10),
        @c_first        char(16),
        @c_middle       char(2),
        @c_street_1     char(20),
        @c_street_2     char(20),
        @c_city         char(20),
        @c_state        char(2),
        @c_zip          char(9),
        @c_phone        char(16),
        @c_since        datetime,
        @c_credit       char(2),
        @c_credit_lim   money,
        @c_balance      money,
        @c_discount     smallmoney,
        @c_data         char(42),
        @datetime       datetime,
        @w_ytd          money,
        @d_ytd          money,
        @cnt            smallint,
        @val            smallint,
        @screen_data    char(200),
        @d_id_local     tinyint,
        @w_id_local     int,
        @c_id_local     int

SELECT  @screen_data    = ""

BEGIN TRANSACTION p
    --  get payment date
    SELECT  @datetime = GETDATE()

    IF (@c_id = 0)
    BEGIN
        --  get customer id and info using last name
        SELECT  @cnt    = COUNT(*)
        FROM    customer WITH (repeatableread)
        WHERE   c_last  = @c_last AND
                          c_w_id  = @c_w_id AND
                          c_d_id  = @c_d_id

        SELECT  @val    = (@cnt + 1) / 2
```

```
    SET     rowcount @val

    SELECT  @c_id   = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_last  = @c_last AND
                        c_w_id  = @c_w_id AND
                        c_d_id  = @c_d_id
    ORDER   BY c_last, c_first

    SET     rowcount 0
END

--  get customer info and update balances
UPDATE  customer
SET     @c_balance      = c_balance = c_balance - @h_amount,
        c_payment_cnt   = c_payment_cnt + 1,
        c_ytd_payment   = c_ytd_payment + @h_amount,
        @c_first        = c_first,
        @c_middle       = c_middle,
        @c_last         = c_last,
        @c_street_1     = c_street_1,
        @c_street_2     = c_street_2,
        @c_city         = c_city,
        @c_state        = c_state,
        @c_zip          = c_zip,
        @c_phone        = c_phone,
        @c_credit       = c_credit,
        @c_credit_lim   = c_credit_lim,
        @c_discount     = c_discount,
        @c_since        = c_since,
        @c_id_local     = c_id
WHERE   c_id            = @c_id AND
        c_w_id          = @c_w_id AND
        c_d_id          = @c_d_id

--  if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    --          compute new info
    SELECT  @c_data =   convert(char(5),@c_id) +
                        convert(char(4),@c_d_id) +
                        convert(char(5),@c_w_id) +
                        convert(char(4),@d_id) +
                        convert(char(5),@w_id) +
                        convert(char(19),@h_amount)

    --  update customer info
    UPDATE  customer
    SET     c_data          = @c_data + substring(c_data, 1, 458),
            @screen_data    = @c_data + substring(c_data, 1, 158)
    WHERE   c_id            = @c_id AND
            c_w_id          = @c_w_id AND
            c_d_id          = @c_d_id
END

--  get district data and update year-to-date
UPDATE  district
SET     d_ytd       = d_ytd + @h_amount,
        @d_street_1 = d_street_1,
        @d_street_2 = d_street_2,
        @d_city     = d_city,
        @d_state    = d_state,
        @d_zip      = d_zip,
        @d_name     = d_name,
        @d_id_local = d_id
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

--  get warehouse data and update year-to-date
UPDATE  warehouse
SET     w_ytd       = w_ytd + @h_amount,
        @w_street_1 = w_street_1,
        @w_street_2 = w_street_2,
        @w_city     = w_city,
        @w_state     = w_state,
        @w_zip      = w_zip,
        @w_name     = w_name,
        @w_id_local = w_id
WHERE   w_id        = @w_id

--  create history record
      INSERT INTO         history VALUES (@c_id_local,
                            @c_d_id,
                            @c_w_id,
                            @d_id_local,
                            @w_id_local,
                            @datetime,
                            @h_amount,
                            @w_name + '    ' + @d_name)

COMMIT TRANSACTION p
```

```
    --  return data to client
    SELECT  @c_id,
            @c_last,
            @datetime,
            @w_street_1,
            @w_street_2,
            @w_city,
            @w_state,
            @w_zip,
            @d_street_1,
            @d_street_2,
            @d_city,
            @d_state,
            @d_zip,
            @c_first,
            @c_middle,
            @c_street_1,
            @c_street_2,
            @c_city,
            @c_state,
            @c_zip,
            @c_phone,
            @c_since,
            @c_credit,
            @c_credit_lim,
            @c_discount,
            @c_balance,
            @screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
----------------------------------------------------------------------
--                                                                  --
--  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 availible order id and update
--  plus initialize local variables
------------------------------------------------------------
    UPDATE  district
    SET     @d_tax          = d_tax,
            @o_id           = d_next_o_id,
            d_next_o_id     = d_next_o_id + 1,
            @o_entry_d      = GETDATE(),
            @li_no          = 0,
            @commit_flag    = 1
    WHERE   d_w_id          = @w_id AND
            d_id            = @d_id

----------------------
--  process orderlines
----------------------
    WHILE (@li_no < @o_ol_cnt)
    BEGIN
        SELECT  @li_no  = @li_no + 1

-----------------------------------------------
--  set i_id, s_w_id, and qty for this lineitem
-----------------------------------------------
        SELECT  @li_id = CASE @li_no
                                WHEN 1 THEN @i_id1
                                WHEN 2 THEN @i_id2
                                WHEN 3 THEN @i_id3
                                WHEN 4 THEN @i_id4
                                WHEN 5 THEN @i_id5
                                WHEN 6 THEN @i_id6
                                WHEN 7 THEN @i_id7
                                WHEN 8 THEN @i_id8
                                WHEN 9 THEN @i_id9
                                WHEN 10 THEN @i_id10
                                WHEN 11 THEN @i_id11
                                WHEN 12 THEN @i_id12
                                WHEN 13 THEN @i_id13
                                WHEN 14 THEN @i_id14
                                WHEN 15 THEN @i_id15
                         END,

                @li_s_w_id = CASE @li_no
                                WHEN 1 THEN @s_w_id1
                                WHEN 2 THEN @s_w_id2
                                WHEN 3 THEN @s_w_id3
                                WHEN 4 THEN @s_w_id4
                                WHEN 5 THEN @s_w_id5
                                WHEN 6 THEN @s_w_id6
                                WHEN 7 THEN @s_w_id7
                                WHEN 8 THEN @s_w_id8
                                WHEN 9 THEN @s_w_id9
                                WHEN 10 THEN @s_w_id10
                                WHEN 11 THEN @s_w_id11
                                WHEN 12 THEN @s_w_id12
                                WHEN 13 THEN @s_w_id13
                                WHEN 14 THEN @s_w_id14
                                WHEN 15 THEN @s_w_id15
                             END,

                @li_qty = CASE  @li_no
                                WHEN 1 THEN @ol_qty1
                                WHEN 2 THEN @ol_qty2
                                WHEN 3 THEN @ol_qty3
                                WHEN 4 THEN @ol_qty4
                                WHEN 5 THEN @ol_qty5
                                WHEN 6 THEN @ol_qty6
                                WHEN 7 THEN @ol_qty7
                                WHEN 8 THEN @ol_qty8
                                WHEN 9 THEN @ol_qty9
                                WHEN 10 THEN @ol_qty10
                                WHEN 11 THEN @ol_qty11
                                WHEN 12 THEN @ol_qty12
                                WHEN 13 THEN @ol_qty13
                                WHEN 14 THEN @ol_qty14
                                WHEN 15 THEN @ol_qty15
                          END
```

```
--------------------------------------
--  get item data (no one updates item)
--------------------------------------
        SELECT  @i_price    = i_price,
                @i_name     = i_name,
                @i_data     = i_data
        FROM    item WITH (repeatableread)
        WHERE   i_id        = @li_id

----------------------
--  update stock values
----------------------
        UPDATE  stock
        SET     s_ytd           = s_ytd + @li_qty,
                @s_quantity     = s_quantity   = s_quantity - @li_qty +
                                    CASE WHEN (s_quantity - @li_qty < 10) THEN 91 ELSE 0 END,
                s_order_cnt     = s_order_cnt + 1,
                s_remote_cnt    = s_remote_cnt +
                                    CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1 END,
                @s_data         = s_data,
                @s_dist         = CASE @d_id
                                    WHEN 1  THEN s_dist_01
                                    WHEN 2  THEN s_dist_02
                                    WHEN 3  THEN s_dist_03
                                    WHEN 4  THEN s_dist_04
                                    WHEN 5  THEN s_dist_05
                                    WHEN 6  THEN s_dist_06
                                    WHEN 7  THEN s_dist_07
                                    WHEN 8  THEN s_dist_08
                                    WHEN 9  THEN s_dist_09
                                    WHEN 10 THEN s_dist_10
                                  END
        WHERE   s_i_id          = @li_id AND
                s_w_id          = @li_s_w_id

-------------------------------------------------------------------------
--  if there actually is a stock (and item) with these ids, go to work
-------------------------------------------------------------------------
        IF (@@rowcount > 0)
        BEGIN
-------------------------------------------------------------
--  insert order_line data (using data from item and stock)
-------------------------------------------------------------
            INSERT INTO order_line VALUES(  @o_id,
                                            @d_id,
                                            @w_id,
                                            @li_no,
                                            @li_id,
                                            'dec 31, 1899',
                                            @i_price * @li_qty,
                                            @li_s_w_id,
                                            @li_qty,
                                            @s_dist)

--------------------------------
--  send line-item data to client
--------------------------------
            SELECT  @i_name,
                    @s_quantity,
                    b_g = CASE  WHEN (   (patindex('%ORIGINAL%',@i_data) > 0) AND
                                                                        (patindex('%ORIGINAL%',@s_data) > 0) )
                                                                THEN 'B' ELSE 'G' END,
                    @i_price,
                    @i_price * @li_qty
                    END
                    ELSE
                    BEGIN
-----------------------------------------------------------
--  no item (or stock) found - triggers rollback condition
-----------------------------------------------------------
            SELECT  '',0,'',0,0
            SELECT  @commit_flag    = 0
                    END
          END
-----------------------------------------------------------
--  get customer last name, discount, and credit rating
-----------------------------------------------------------
    SELECT  @c_last     = c_last,
            @c_discount = c_discount,
            @c_credit   = c_credit,
            @c_id_local = c_id
    FROM    customer WITH (repeatableread)
    WHERE   c_id        = @c_id AND
            c_w_id      = @w_id AND
            c_d_id      = @d_id

---------------------------------------
--  insert fresh row into orders table
---------------------------------------
    INSERT INTO orders VALUES ( @o_id,
                                @d_id,
                                @w_id,
                                @c_id_local,
```

```
                                0,
                                @o_ol_cnt,
                                @o_all_local,
                                @o_entry_d)

-------------------------------------------------
--  insert corresponding row into new-order table
-------------------------------------------------
    INSERT INTO new_order VALUES (  @o_id,
                                    @d_id,
                                    @w_id)

------------------------
--  select warehouse tax
------------------------
    SELECT  @w_tax  = w_tax
    FROM    warehouse WITH (repeatableread)
    WHERE   w_id    = @w_id

    IF (@commit_flag = 1)
                    COMMIT TRANSACTION n
    ELSE
------------------------------
--  all that work for nuthin!!!
------------------------------
        ROLLBACK TRANSACTION n

------------------------------
--  return order data to client
------------------------------
    SELECT  @w_tax,
            @d_tax,
            @o_id,
            @c_last,
            @c_discount,
            @c_credit,
            @o_entry_d,
            @commit_flag
END
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
----------------------------------------------------------------------
--                                                                  --
--  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 updlock)
        WHERE   no_w_id = @w_id AND
                no_d_id = @d_id
        ORDER   BY no_o_id ASC

        IF (@@rowcount <> 0)
        BEGIN
            --  claim the order for this district
            DELETE  new_order
            WHERE   no_w_id = @w_id AND
                    no_d_id = @d_id AND
                    no_o_id = @o_id

            --  set carrier_id on this order (and get customer id)
            UPDATE  orders
            SET     o_carrier_id    = @o_carrier_id,
                    @c_id           = o_c_id
            WHERE   o_w_id          = @w_id AND
                    o_d_id          = @d_id AND
                    o_id            = @o_id

            --  set date in all lineitems for this order (and sum amounts)
            UPDATE  order_line
            SET     ol_delivery_d   = GETDATE(),
                    @total          = @total + ol_amount
            WHERE   ol_w_id         = @w_id AND
                    ol_d_id         = @d_id AND
                    ol_o_id         = @o_id

            --  accummulate lineitem amounts for this order into customer
            UPDATE  customer
            SET     c_balance       = c_balance + @total,
                    c_delivery_cnt  = c_delivery_cnt + 1
            WHERE   c_w_id          = @w_id AND
                    c_d_id          = @d_id AND
                    c_id            = @c_id
        END

    SELECT  @oid1   = CASE @d_id WHEN   1   THEN @o_id ELSE @oid1   END,
            @oid2   = CASE @d_id WHEN   2   THEN @o_id ELSE @oid2   END,
            @oid3   = CASE @d_id WHEN   3   THEN @o_id ELSE @oid3   END,
            @oid4   = CASE @d_id WHEN   4   THEN @o_id ELSE @oid4   END,
            @oid5   = CASE @d_id WHEN   5   THEN @o_id ELSE @oid5   END,
            @oid6   = CASE @d_id WHEN   6   THEN @o_id ELSE @oid6   END,
            @oid7   = CASE @d_id WHEN   7   THEN @o_id ELSE @oid7   END,
            @oid8   = CASE @d_id WHEN   8   THEN @o_id ELSE @oid8   END,
            @oid9   = CASE @d_id WHEN   9   THEN @o_id ELSE @oid9   END,
            @oid10  = CASE @d_id WHEN  10   THEN @o_id ELSE @oid10  END
    END

COMMIT TRANSACTION d

-- return delivery data to client

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

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO
----------------------------------------------------------------------
--                                                                  --
--  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,
                        @threshhold     smallint

AS
DECLARE @o_id_low   int,
        @o_id_high  int

SELECT  @o_id_low   = (d_next_o_id - 20),
        @o_id_high  = (d_next_o_id - 1)
FROM    district
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

SELECT  COUNT(DISTINCT(s_i_id))
FROM    stock,
        order_line
WHERE   ol_w_id     = @w_id AND
        ol_d_id     = @d_id and
        ol_o_id     BETWEEN @o_id_low AND
                            @o_id_high AND
        s_w_id      = ol_w_id AND
        s_i_id      = ol_i_id AND
        s_quantity  < @threshhold
OPTION(ORDER GROUP)
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

//        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 tables
          BOOL                                  table_item;                       // set if loading ITEM table
specifically
          BOOL                                  table_warehouse;    // set if loading WAREHOUSE, DISTRICT, and STOCK
          BOOL                                  table_customer;              // set if loading CUSTOMER and HISTORY
          BOOL                                  table_orders;                // set if loading NEW-ORDER, ORDERS,
ORDER-LINE
    long                                        num_warehouses;
    long                                        batch;
    long                                        verbose;
          long                                  pack_size;
          char                                  *loader_res_file;
          char                                  *log_path;
          char                                  *synch_servername;
          long                                  case_sensitivity;
          long                                  starting_warehouse;
          long                                  build_index;
          long                                  index_order;
          long                                  scale_down;
          char                                  *index_script_path;
} 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();
```

```
//=====================================================================
//        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;
    long                ol_i_id;
          long                                  ol_supply_w_id;
    short               ol_quantity;
    double              ol_amount;
    char                ol_dist_info[DIST_INFO_LEN+1];
          char                                  ol_delivery_d[OL_DELIVERY_D_LEN+1];
} ORDER_LINE_STRUCT;

typedef struct
{
    long                o_id;
    short               o_d_id;
          long                                  o_w_id;
    long                o_c_id;
    short               o_carrier_id;
    short               o_ol_cnt;
    short               o_all_local;
          ORDER_LINE_STRUCT   o_ol[15];
} ORDERS_STRUCT;

typedef struct
{
    long                                c_id;
    short                               c_d_id;
          long                                  c_w_id;
          char                                  c_first[FIRST_NAME_LEN+1];
    char                                c_middle[MIDDLE_NAME_LEN+1];
    char                                c_last[LAST_NAME_LEN+1];
    char                                c_street_1[ADDRESS_LEN+1];
    char                                c_street_2[ADDRESS_LEN+1];
    char                                c_city[ADDRESS_LEN+1];
    char                                c_state[STATE_LEN+1];
    char                                c_zip[ZIP_LEN+1];
    char                                c_phone[PHONE_LEN+1];
    char                                c_credit[CREDIT_LEN+1];
    double                                      c_credit_lim;
    double                                      c_discount;
          char                                  c_balance[6];
    double                                      c_ytd_payment;
          short                                 c_payment_cnt;
```

```
          short                                   c_delivery_cnt;
          char                                    c_data[C_DATA_LEN+1];
          double                                  h_amount;
    char                                  h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
          char                                    c_last[LAST_NAME_LEN+1];
          char                                    c_first[FIRST_NAME_LEN+1];
    long                                  c_id;
} CUSTOMER_SORT_STRUCT;

typedef struct
{
    long                time_start;
} LOADER_TIME_STRUCT;


// Global variables
char      szLastError[300];

HENV      henv;

HDBC      v_hdbc;                                                   // for SQL Server version verification
HDBC      i_hdbc1;                                        // for ITEM table
HDBC      w_hdbc1;                                        // for WAREHOUSE, DISTRICT, STOCK
HDBC      c_hdbc1;                                        // for CUSTOMER
HDBC      c_hdbc2;                                        // for HISTORY
HDBC      o_hdbc1;                                        // for ORDERS
HDBC      o_hdbc2;                                        // for NEW-ORDER
HDBC      o_hdbc3;                                        // for ORDER-LINE

HSTMT     v_hstmt;                                        // for SQL Server version verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

int                total_db_errors;

ORDERS_STRUCT       orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT     customer_buf[CUSTOMERS_PER_DISTRICT];
long                          orders_rows_loaded;
double                        new_order_rows_loaded;
double                        order_line_rows_loaded;
long                          history_rows_loaded;
long                          customer_rows_loaded;
double                        stock_rows_loaded;
long                          district_rows_loaded;
long                          item_rows_loaded;
long                          warehouse_rows_loaded;
long                          main_time_start;
long                          main_time_end;
long                          max_items;
long                          customers_per_district;
long                          orders_per_district;
long                          first_new_order;
long                          last_new_order;

TPCCLDR_ARGS    *aptr, args;

//====================================================================
//
// Function name: main
//
//====================================================================
int main(int  argc, char **argv)
{

          DWORD           dwThreadID[MAX_MAIN_THREADS];
          HANDLE          hThread[MAX_MAIN_THREADS];
          FILE            *fLoader;
          char            buffer[255];
          int                                   i;

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

    printf("\n*************************************************");
          printf("\n*                                               *");
          printf("\n*  Microsoft SQL Server                         *");
          printf("\n*                                               *");
          printf("\n*  TPC-C BENCHMARK KIT:  Database loader        *");
          printf("\n*  Version %s                                 *", TPCKIT_VER);
          printf("\n*                                               *");
          printf("\n*************************************************\n\n");

    // process command line arguments
    aptr = &args;
    GetArgsLoader(argc, argv, aptr);

          printf("Build interface is ODBC.\n");
```

```
        if (aptr->build_index == 0)
                printf("Data load only - no index creation.\n");
        else
                printf("Data load and index creation.\n");

        if (aptr->index_order == 0)
                printf("Clustered indexes will be created after bulk load.\n");
        else
                printf("Clustered indexes will be created before bulk load.\n");

        // set database scale values
        if (aptr->scale_down == 1)
        {
                printf("*** Scaled Down Database ***\n");
                max_items = MAXITEMS_SCALE_DOWN;
                customers_per_district = CUSTOMERS_SCALE_DOWN;
                orders_per_district = ORDERS_SCALE_DOWN;
                first_new_order = 0;
                last_new_order  = 30;
        }
        else
        {
                max_items = MAXITEMS;
                customers_per_district = CUSTOMERS_PER_DISTRICT;
                orders_per_district = ORDERS_PER_DISTRICT;
                first_new_order = 2100;
                last_new_order  = 3000;
        }

        // open connections to SQL Server
        OpenConnections();

        // open file for loader results
        fLoader = fopen(aptr->loader_res_file, "w");


    if (fLoader == NULL)
          {
                printf("Error, loader result file open failed.");
                exit(-1);
          }

    // start loading data
    sprintf(buffer,"TPC-C load started for %ld warehouses.\n",aptr->num_warehouses);
          if        (aptr->scale_down == 1)
          {
                sprintf(buffer,"SCALED DOWN DATABASE.\n");
          }

          printf("%s",buffer);
          fprintf(fLoader,"%s",buffer);

          main_time_start = (TimeNow() / MILLI);

          // start parallel load threads
          if (aptr->tables_all || aptr->table_item)
          {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,
                                                                        0,
                                                                        (LPTHREAD_START_ROUTINE) LoadItem,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread = 0.\n");
                        exit(-1);
                }
          }

          if (aptr->tables_all || aptr->table_warehouse)
          {
                fprintf(fLoader, "Starting loader threads for: warehouse\n");

                hThread[1] = CreateThread(NULL,
                                                                        0,
                                                                        (LPTHREAD_START_ROUTINE)
LoadWarehouse,
                                                                        NULL,
                                                                        0,
                                                                        &dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread = 1.\n");
                        exit(-1);
                }
          }
```

```
          if (aptr->tables_all || aptr->table_customer)
          {
                    fprintf(fLoader, "Starting loader threads for: customer\n");

                    hThread[2] = CreateThread(NULL,
                                                                                0,
                                                                                (LPTHREAD_START_ROUTINE)
LoadCustomer,
                                                                                NULL,
                                                                                0,
                                                                                &dwThreadID[2]);

                    if (hThread[2] == NULL)
                    {
                              printf("Error, failed in creating creating main thread = 2.\n");
                              exit(-1);
                    }
          }

          if (aptr->tables_all || aptr->table_orders)
          {
                    fprintf(fLoader, "Starting loader threads for: orders\n");

                    hThread[3] = CreateThread(NULL,
                                                                                0,
                                                                                (LPTHREAD_START_ROUTINE) LoadOrders,
                                                                                NULL,
                                                                                0,
                                                                                &dwThreadID[3]);

                    if (hThread[3] == NULL)
                    {
                              printf("Error, failed in creating creating main thread = 3.\n");
                              exit(-1);
                    }
          }

          // Wait for threads to finish...
          for (i=0; i<MAX_MAIN_THREADS; i++)
          {
                    if (hThread[i] != NULL)
                    {
                              WaitForSingleObject( hThread[i], INFINITE );
                              CloseHandle(hThread[i]);
                              hThread[i] = NULL;
                    }
          }

          main_time_end = (TimeNow() / MILLI);

    sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                              (main_time_end - main_time_start)/60);

          printf("%s",buffer);
          fprintf(fLoader, "%s", buffer);

          fclose(fLoader);

          SQLFreeEnv(henv);

    exit(0);

          return 0;
}


//===============================================================================
//
// Function name: LoadItem
//
//===============================================================================
void LoadItem()
{
    int                       i;
          long                i_id;
          long                i_im_id;
    char              i_name[I_NAME_LEN+1];
    double                    i_price;
    char              i_data[I_DATA_LEN+1];
          char                name[20];
          long                time_start;
          RETCODE             rc;
          DBINT               rcint;
          char                bcphint[128];
          char                err_log_path[256];

          // Seed with unique number
          seed(11);

          printf("Loading item table...\n");

          //if build index before load
          if ((aptr->build_index == 1) && (aptr->index_order == 1))
```

```
                BuildIndex("idxitmcl");

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

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

        strcpy(err_log_path,aptr->log_path);
        strcat(err_log_path,"item.err");
        rc = bcp_init(i_hdbc1, name, NULL, err_log_path , DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (i_id), ROWS_PER_BATCH = 100000");
                rc = bcp_control(i_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);
        }

        i = 0;
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_name, 0, I_NAME_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_price, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) i_data, 0, SQL_VARLEN_DATA, "", 1, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);
        rc = bcp_bind(i_hdbc1, (BYTE *) &i_im_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(i_hdbc1);

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

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

                MakeAlphaStringPadded(14, 24, I_NAME_LEN, i_name);

                i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

                rc = bcp_sendrow(i_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                item_rows_loaded++;
                CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item", &time_start);
        }

        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}

//============================================================================
//
// Function   : LoadWarehouse
//
// Loads WAREHOUSE table and loads Stock and District as Warehouses are created
//
//============================================================================
void LoadWarehouse()
{
        int             i;
        long    w_id;
    char  w_name[W_NAME_LEN+1];
    char  w_street_1[ADDRESS_LEN+1];
    char  w_street_2[ADDRESS_LEN+1];
    char  w_city[ADDRESS_LEN+1];
    char  w_state[STATE_LEN+1];
    char  w_zip[ZIP_LEN+1];
```

```
double          w_tax;
double          w_ytd;
      char      name[20];
      long      time_start;
      RETCODE   rc;
      DBINT     rcint;
      char      bcphint[128];
      char      err_log_path[256];

      // Seed with unique number
      seed(2);

      printf("Loading warehouse table...\n");

      // if build index before load...
      if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxwarcl");

      InitString(w_name, W_NAME_LEN+1);
      InitAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

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

      strcpy(err_log_path,aptr->log_path);
      strcat(err_log_path,"whouse.err");
      rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);

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

      if ((aptr->build_index == 1) && (aptr->index_order == 1))
      {
                sprintf(bcphint, "tablock, order (w_id), ROWS_PER_BATCH = %d", aptr->num_warehouses);
                rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                          HandleErrorDBC(w_hdbc1);
      }

      i = 0;
      rc = bcp_bind(w_hdbc1, (BYTE *) &w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
      if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
      rc = bcp_bind(w_hdbc1, (BYTE *) &w_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
      if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
      rc = bcp_bind(w_hdbc1, (BYTE *) &w_tax, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
      if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
      rc = bcp_bind(w_hdbc1, (BYTE *) w_name, 0, W_NAME_LEN, NULL, 0, 0, ++i);
      if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);
      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;
          char      name[20];
    long  d_next_o_id;
          long      time_start;
          long      w_id;
          RETCODE   rc;
          DBINT     rcint;
          char      bcphint[128];
          char      err_log_path[256];

          // Seed with unique number
          seed(4);

          printf("Loading district table...\n");

          // build index before load
          if ((aptr->build_index == 1) && (aptr->index_order == 1))
                    BuildIndex("idxdiscl");

          InitString(d_name, D_NAME_LEN+1);
          InitAddress(d_street_1, d_street_2, d_city, d_state, d_zip);
          sprintf(name, "%s..%s", aptr->database, "district");

          strcpy(err_log_path,aptr->log_path);
          strcat(err_log_path,"district.err");
          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 != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) d_state, 0, STATE_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) d_zip, 0, ZIP_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          d_ytd  = 30000.0;

          d_next_o_id = orders_per_district+1;

          time_start = (TimeNow() / MILLI);

          for (w_id = aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
          {
                    d_w_id = w_id;

                    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                    {
                              MakeAlphaStringPadded(6,10,D_NAME_LEN, d_name);

                              MakeAddress(d_street_1, d_street_2, d_city, d_state, d_zip);

                              d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                              rc = bcp_sendrow(w_hdbc1);
                              if (rc != SUCCEED)
                                        HandleErrorDBC(w_hdbc1);

                              district_rows_loaded++;
                              CheckForCommit(w_hdbc1, w_hstmt1, district_rows_loaded, "district", &time_start);
                    }
    }

          rcint = bcp_done(w_hdbc1);
          if (rcint < 0)
                    HandleErrorDBC(w_hdbc1);

          printf("Finished loading district table.\n");

          // if build index after load...
          if ((aptr->build_index == 1) && (aptr->index_order == 0))
                    BuildIndex("idxdiscl");

    return;
}

//============================================================================
//
// Function   : Stock
//
//============================================================================
void Stock()
{
          int                i;
          long      s_i_id;
          long      s_w_id;
    short s_quantity;
    char  s_dist_01[S_DIST_LEN+1];
    char  s_dist_02[S_DIST_LEN+1];
    char  s_dist_03[S_DIST_LEN+1];
    char  s_dist_04[S_DIST_LEN+1];
    char  s_dist_05[S_DIST_LEN+1];
    char  s_dist_06[S_DIST_LEN+1];
    char  s_dist_07[S_DIST_LEN+1];
    char  s_dist_08[S_DIST_LEN+1];
    char  s_dist_09[S_DIST_LEN+1];
    char  s_dist_10[S_DIST_LEN+1];
    long  s_ytd;
    short s_order_cnt;
    short s_remote_cnt;
    char  s_data[S_DATA_LEN+1];
    short len;
          char      name[20];
          long      time_start;
          RETCODE   rc;
          DBINT     rcint;
          char      bcphint[128];
          char      err_log_path[256];

          // Seed with unique number
          seed(3);

          // if build index before load...
          if ((aptr->build_index == 1) && (aptr->index_order == 1))
                    BuildIndex("idxstkcl");

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

          strcpy(err_log_path,aptr->log_path);
```

```
          strcat(err_log_path,"stock.err");
          rc = bcp_init(w_hdbc1, name, NULL, err_log_path, DB_IN);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          if ((aptr->build_index == 1) && (aptr->index_order == 1))
          {
                    sprintf(bcphint, "tablock, order (s_i_id, s_w_id), ROWS_PER_BATCH = %u", (aptr->num_warehouses *
100000));
                    rc = bcp_control(w_hdbc1, BCPHINTS, (void*) bcphint);
                    if (rc != SUCCEED)
                              HandleErrorDBC(w_hdbc1);
          }

          i = 0;
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_ytd, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_order_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) &s_remote_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, SQL_VARLEN_DATA, "", 1, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_01, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_02, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_03, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_04, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_05, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_06, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_07, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_09, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
          rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_10, 0, S_DIST_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

          s_ytd = s_order_cnt = s_remote_cnt = 0;

          time_start = (TimeNow() / MILLI);

          printf("...Loading stock table\n");

          for (s_i_id=1; s_i_id <= max_items; s_i_id++)
          {

                    for (s_w_id = (long)aptr->starting_warehouse; s_w_id <= aptr->num_warehouses; s_w_id++)
                    {

                              s_quantity = (short)RandomNumber(10L,100L);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_01);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_02);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_03);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_04);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_05);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_06);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_07);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_08);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_09);
                              len = MakeAlphaString(24,24,S_DIST_LEN, s_dist_10);

                              len = MakeOriginalAlphaString(26,50, S_DATA_LEN, s_data,10);
```

```
                        rc = bcp_sendrow(w_hdbc1);
                        if (rc != SUCCEED)
                                HandleErrorDBC(w_hdbc1);

                        stock_rows_loaded++;
                        CheckForCommit_Big(w_hdbc1, w_hstmt1, stock_rows_loaded, "stock", &time_start);
                }
        }

        rcint = bcp_done(w_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(w_hdbc1);

        printf("Finished loading stock table.\n");

        SQLFreeStmt(w_hstmt1, SQL_DROP);
        SQLDisconnect(w_hdbc1);
        SQLFreeConnect(w_hdbc1);

        // if build index after load...
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxstkcl");

        return;
}

//====================================================================
//
// Function   : LoadCustomer
//
//====================================================================
void LoadCustomer()
{
        LOADER_TIME_STRUCT      customer_time_start;
        LOADER_TIME_STRUCT      history_time_start;
        long                                            w_id;
    short                       d_id;
        DWORD                           dwThreadID[MAX_CUSTOMER_THREADS];
        HANDLE                          hThread[MAX_CUSTOMER_THREADS];
        char                            name[20];
        RETCODE                                         rc;
        DBINT                                           rcint;
        char                                            bcphint[128];
        char                                            cmd[256];
        int                                                     num_procs;
        char                                            err_log_path_cust[256];
        char                                            err_log_path_hist[256];

        // Seed with unique number
        seed(5);

        printf("Loading customer and history tables...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                BuildIndex("idxcuscl");
                 // check the number of processors on this system
                 // if 8 or more processors, then build index on History.
                 // if less than 8 processors, do not build the index
                num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
                if ( num_procs >= 8 )
                        BuildIndex("idxhiscl");
        }

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "customer");

        strcpy(err_log_path_cust,aptr->log_path);
        strcat(err_log_path_cust,"customer.err");
        rc = bcp_init(c_hdbc1, name, NULL, err_log_path_cust, DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                sprintf(bcphint, "tablock, order (c_w_id, c_d_id, c_id), ROWS_PER_BATCH = %u", (aptr-
>num_warehouses * 30000));
                rc = bcp_control(c_hdbc1, BCPHINTS, (void*) bcphint);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc1);
        }

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

        rc = bcp_init(c_hdbc2, name, NULL, "logs\\history.err", DB_IN);
        strcpy(err_log_path_hist,aptr->log_path);
        strcat(err_log_path_hist,"history.err");
        rc = bcp_init(c_hdbc2, name, NULL, err_log_path_hist, DB_IN);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

        sprintf(bcphint, "tablock");
```

```
	rc = bcp_control(c_hdbc2, BCPHINTS, (void*) bcphint);
	if (rc != SUCCEED)
		HandleErrorDBC(c_hdbc2);

	customer_rows_loaded    = 0;
	history_rows_loaded     = 0;

	CustomerBufInit();

	customer_time_start.time_start = (TimeNow() / MILLI);
	history_time_start.time_start = (TimeNow() / MILLI);

	for (w_id = (long)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
	{
		for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
		{

			CustomerBufLoad(d_id, w_id);

			// Start parallel loading threads here...
			// Start customer table thread
			printf("...Loading customer table for: d_id = %d, w_id = %d\n", d_id, w_id);

			hThread[0] = CreateThread(NULL,
                                                                        0,
                                                                        (LPTHREAD_START_ROUTINE)
LoadCustomerTable,
                                                                        &customer_time_start,
                                                                        0,
                                                                        &dwThreadID[0]);

			if (hThread[0] == NULL)
			{
				printf("Error, failed in creating creating thread = 0.\n");
				exit(-1);
			}

			// Start History table thread
			printf("...Loading history table for: d_id = %d, w_id = %d\n", d_id, w_id);

			hThread[1] = CreateThread(NULL,
                                                                        0,
                                                                        (LPTHREAD_START_ROUTINE)
LoadHistoryTable,
                                                                        &history_time_start,
                                                                        0,
                                                                        &dwThreadID[1]);

			if (hThread[1] == NULL)
			{
				printf("Error, failed in creating creating thread = 1.\n");
				exit(-1);
			}

			WaitForSingleObject( hThread[0], INFINITE );
			WaitForSingleObject( hThread[1], INFINITE );

			if (CloseHandle(hThread[0]) == FALSE)
			{
				printf("Error, failed in closing customer thread handle with errno: %d\n",
GetLastError());
			}

			if (CloseHandle(hThread[1]) == FALSE)
			{
				printf("Error, failed in closing history thread handle with errno: %d\n",
GetLastError());
			}
		}
	}

	// flush the bulk connection
	rcint = bcp_done(c_hdbc1);
	if (rcint < 0)
		HandleErrorDBC(c_hdbc1);

	rcint = bcp_done(c_hdbc2);
	if (rcint < 0)
		HandleErrorDBC(c_hdbc2);

	printf("Finished loading customer table.\n");

	// if build index after load...
	if ((aptr->build_index == 1) && (aptr->index_order == 0))
	{
		BuildIndex("idxcuscl");
		// check the number of processors on this system
		// if 8 or more processors, then build index on History.
		// if less than 8 processors, do not build the index
		num_procs = atoi(getenv( "NUMBER_OF_PROCESSORS" ));
		if (num_procs >= 8)
			BuildIndex("idxhiscl");
	}
```

```
          // build non-clustered index
          if (aptr->build_index == 1)
                    BuildIndex("idxcusnc");

          // Output the NURAND used for the loader into C_FIRST for C_ID = 1,
          // C_W_ID = 1, and C_D_ID = 1
          sprintf(cmd, "osql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first = 'C_LOAD = %d' where c_id = 1 and
c_w_id = 1 and c_d_id = 1\" > %snurand_load.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr->database,
                                        LOADER_NURAND_C,
                                        aptr->log_path);

          system(cmd);

          SQLFreeStmt(c_hstmt1, SQL_DROP);
          SQLDisconnect(c_hdbc1);
          SQLFreeConnect(c_hdbc1);

          SQLFreeStmt(c_hstmt2, SQL_DROP);
          SQLDisconnect(c_hdbc2);
          SQLFreeConnect(c_hdbc2);

    return;
}

//====================================================================
//
// Function   : CustomerBufInit
//
//====================================================================
void CustomerBufInit()
{
          long     i;

          for (i=0;i<customers_per_district;i++)
          {
                    customer_buf[i].c_id = 0;
                    customer_buf[i].c_d_id = 0;
                    customer_buf[i].c_w_id = 0;

                    strcpy(customer_buf[i].c_first,"");
                    strcpy(customer_buf[i].c_middle,"");
                    strcpy(customer_buf[i].c_last,"");
                    strcpy(customer_buf[i].c_street_1,"");
                    strcpy(customer_buf[i].c_street_2,"");
                    strcpy(customer_buf[i].c_city,"");
                    strcpy(customer_buf[i].c_state,"");
                    strcpy(customer_buf[i].c_zip,"");
                    strcpy(customer_buf[i].c_phone,"");
                    strcpy(customer_buf[i].c_credit,"");

                    customer_buf[i].c_credit_lim = 0;
                    customer_buf[i].c_discount = (float) 0;

                    strcpy(customer_buf[i].c_balance,"");

                    customer_buf[i].c_ytd_payment = 0;
                    customer_buf[i].c_payment_cnt = 0;
                    customer_buf[i].c_delivery_cnt = 0;

                    strcpy(customer_buf[i].c_data,"");

                    customer_buf[i].h_amount = 0;

                    strcpy(customer_buf[i].h_data,"");
          }
}

//====================================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//====================================================================
void CustomerBufLoad(int d_id, long w_id)
{
          long                                   i;
    CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];

          for (i=0;i<customers_per_district;i++)
          {
                    if (i < 1000)
                              LastName(i, c[i].c_last);
                    else
                              LastName(NURand(255,0,999,LOADER_NURAND_C), c[i].c_last);

                    MakeAlphaStringPadded(8,16,FIRST_NAME_LEN, c[i].c_first);

                    c[i].c_id = i+1;
```

```
		}

		printf("...Loading customer buffer for: d_id = %d, w_id = %d\n",
				d_id, w_id);

		for (i=0;i<customers_per_district;i++)
		{
			customer_buf[i].c_d_id = d_id;
			customer_buf[i].c_w_id = w_id;
			customer_buf[i].h_amount = 10.0;
			customer_buf[i].c_ytd_payment  = 10.0;
			customer_buf[i].c_payment_cnt  = 1;
			customer_buf[i].c_delivery_cnt = 0;
			customer_buf[i].c_id = c[i].c_id;
			strcpy(customer_buf[i].c_first, c[i].c_first);
			strcpy(customer_buf[i].c_last, c[i].c_last);
			customer_buf[i].c_middle[0] = 'O';
			customer_buf[i].c_middle[1] = 'E';
			MakeAddress(customer_buf[i].c_street_1,
					customer_buf[i].c_street_2,
					customer_buf[i].c_city,
					customer_buf[i].c_state,
					customer_buf[i].c_zip);
			MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

			if (RandomNumber(1L, 100L) > 10)
				customer_buf[i].c_credit[0] = 'G';
			else
				customer_buf[i].c_credit[0] = 'B';
			customer_buf[i].c_credit[1] = 'C';
			customer_buf[i].c_credit_lim = 50000.0;
			customer_buf[i].c_discount = ((float) RandomNumber(0L, 5000L)) / 10000.0;

			strcpy(customer_buf[i].c_balance,"-10.0");
			MakeAlphaStringPadded(300, 500, C_DATA_LEN, customer_buf[i].c_data);

			// Generate HISTORY data
			MakeAlphaStringPadded(12, 24, H_DATA_LEN, customer_buf[i].h_data);
		}
}

//==================================================================
//
// Function   : LoadCustomerTable
//
//==================================================================
void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
		long		i;
    long        c_id;
    short       c_d_id;
		long		c_w_id;
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN+1];
    char        c_last[LAST_NAME_LEN+1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;
		char		c_balance[6];
    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
		char		c_since[C_SINCE_LEN+1];
		RETCODE		rc;

    i = 0;
		rc = bcp_bind(c_hdbc1, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
		rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_discount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_credit_lim, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_last, 0, LAST_NAME_LEN, NULL, 0, 0, ++i);
		if (rc != SUCCEED)
			HandleErrorDBC(c_hdbc1);
		rc = bcp_bind(c_hdbc1, (BYTE *) c_first, 0, FIRST_NAME_LEN, NULL, 0, 0, ++i);
		if (rc != SUCCEED)
```

```
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_credit, 0, CREDIT_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_balance, 0, 5, NULL, 0, SQLCHARACTER, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_ytd_payment, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_payment_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) &c_delivery_cnt,0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_1, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_street_2, 0, ADDRESS_LEN, NULL,0,0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_city, 0, ADDRESS_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_state, 0, STATE_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_zip, 0, ZIP_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_phone, 0, PHONE_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
          rc = bcp_bind(c_hdbc1, (BYTE *) &c_since, 0, C_SINCE_LEN, NULL, 0, SQLCHARACTER, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
          rc = bcp_bind(c_hdbc1, (BYTE *) c_middle, 0, MIDDLE_NAME_LEN,NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);
    rc = bcp_bind(c_hdbc1, (BYTE *) c_data, 0, C_DATA_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc1);

    for (i = 0; i < customers_per_district; i++)
    {
                c_id = customer_buf[i].c_id;
                c_d_id = customer_buf[i].c_d_id;
                c_w_id = customer_buf[i].c_w_id;

                strcpy(c_first, customer_buf[i].c_first);
                strcpy(c_middle, customer_buf[i].c_middle);
                strcpy(c_last, customer_buf[i].c_last);
                strcpy(c_street_1, customer_buf[i].c_street_1);
                strcpy(c_street_2, customer_buf[i].c_street_2);
                strcpy(c_city, customer_buf[i].c_city);
                strcpy(c_state, customer_buf[i].c_state);
                strcpy(c_zip, customer_buf[i].c_zip);
                strcpy(c_phone, customer_buf[i].c_phone);
                strcpy(c_credit, customer_buf[i].c_credit);

                FormatDate(&c_since);

                c_credit_lim = customer_buf[i].c_credit_lim;
                c_discount = customer_buf[i].c_discount;
                strcpy(c_balance, customer_buf[i].c_balance);
                c_ytd_payment = customer_buf[i].c_ytd_payment;
                c_payment_cnt = customer_buf[i].c_payment_cnt;
                c_delivery_cnt = customer_buf[i].c_delivery_cnt;
                strcpy(c_data, customer_buf[i].c_data);

                // Send data to server
                rc = bcp_sendrow(c_hdbc1);
                if (rc != SUCCEED)
                          HandleErrorDBC(c_hdbc1);

                customer_rows_loaded++;
                CheckForCommit(c_hdbc1, c_hstmt1, customer_rows_loaded, "customer", &customer_time_start-
>time_start);
    }
}

//======================================================================
//
// Function   : LoadHistoryTable
//
//======================================================================
void LoadHistoryTable(LOADER_TIME_STRUCT *history_time_start)
{
          long                i;
    long        c_id;
    short       c_d_id;
          long                c_w_id;
```

```
        double    h_amount;
    char        h_data[H_DATA_LEN+1];
        char                h_date[H_DATE_LEN+1];
        RETCODE             rc;

        i = 0;
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &c_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
        rc = bcp_bind(c_hdbc2, (BYTE *) &h_date, 0, H_DATE_LEN, NULL, 0, SQLCHARACTER, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) &h_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, ++i);
        if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);

    for (i = 0; i < customers_per_district; i++)
    {
                c_id = customer_buf[i].c_id;
                c_d_id = customer_buf[i].c_d_id;
                c_w_id = customer_buf[i].c_w_id;
                h_amount = customer_buf[i].h_amount;
                strcpy(h_data, customer_buf[i].h_data);

                FormatDate(&h_date);

                // send to server
                rc = bcp_sendrow(c_hdbc2);
                if (rc != SUCCEED)
                        HandleErrorDBC(c_hdbc2);

                history_rows_loaded++;
                CheckForCommit(c_hdbc2, c_hstmt2, history_rows_loaded, "history", &history_time_start->time_start);
    }
}

//============================================================-===============
//
// Function   : LoadOrders
//
//=============================================================================
void LoadOrders()
{
        LOADER_TIME_STRUCT      orders_time_start;
        LOADER_TIME_STRUCT      new_order_time_start;
        LOADER_TIME_STRUCT      order_line_time_start;
        long                                            w_id;
    short                   d_id;
        DWORD                           dwThreadID[MAX_ORDER_THREADS];
        HANDLE                          hThread[MAX_ORDER_THREADS];
        char                            name[20];
        RETCODE                                         rc;
        char                                            bcphint[128];
        char                                            err_log_path_ord[256];
        char                                            err_log_path_nord[256];
        char                                            err_log_path_ordl[256];

        // seed with unique number
        seed(6);

        printf("Loading orders...\n");

        // if build index before load...
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
        {
                        BuildIndex("idxordcl");
                        BuildIndex("idxnodcl");
                        BuildIndex("idxodlcl");
        }

        // initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "orders");

        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 * 300000));
                    rc = bcp_control(o_hdbc3, BCPHINTS, (void*) bcphint);
                    if (rc != SUCCEED)
                              HandleErrorDBC(o_hdbc3);
          }

          orders_rows_loaded      = 0;
          new_order_rows_loaded   = 0;
          order_line_rows_loaded  = 0;

          OrdersBufInit();

          orders_time_start.time_start = (TimeNow() / MILLI);
          new_order_time_start.time_start = (TimeNow() / MILLI);
          order_line_time_start.time_start = (TimeNow() / MILLI);

          for (w_id = (long)aptr->starting_warehouse; w_id <= aptr->num_warehouses; w_id++)
          {
                    for (d_id = 1; d_id <= DISTRICT_PER_WAREHOUSE; d_id++)
                    {

                              OrdersBufLoad(d_id, w_id);

                              // start parallel loading threads here...
                              // start Orders table thread
                              printf("...Loading Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                              hThread[0] = CreateThread(NULL,
                                                                                  0,
                                                                                  (LPTHREAD_START_ROUTINE)
LoadOrdersTable,
                                                                                  &orders_time_start,
                                                                                  0,
                                                                                  &dwThreadID[0]);

                              if (hThread[0] == NULL)
                              {
                                        printf("Error, failed in creating creating thread = 0.\n");
                                        exit(-1);
                              }

                              // start NewOrder table thread
                              printf("...Loading New-Order Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                              hThread[1] = CreateThread(NULL,
                                                                                  0,
                                                                                  (LPTHREAD_START_ROUTINE)
LoadNewOrderTable,
                                                                                  &new_order_time_start,
                                                                                  0,
                                                                                  &dwThreadID[1]);
```

```
                            if (hThread[1] == NULL)
                            {
                                     printf("Error, failed in creating creating thread = 1.\n");
                                     exit(-1);
                            }

                            // start Order-Line table thread
                            printf("...Loading Order-Line Table for: d_id = %d, w_id = %d\n", d_id, w_id);

                            hThread[2] = CreateThread(NULL,
                                                                                                    0,
                                                                                                    (LPTHREAD_START_ROUTINE)
LoadOrderLineTable,
                                                                                                    &order_line_time_start,
                                                                                                    0,
                                                                                                    &dwThreadID[2]);

                            if (hThread[2] == NULL)
                            {
                                     printf("Error, failed in creating creating thread = 2.\n");
                                     exit(-1);
                            }

                            WaitForSingleObject( hThread[0], INFINITE );
                            WaitForSingleObject( hThread[1], INFINITE );
                            WaitForSingleObject( hThread[2], INFINITE );

                            if (CloseHandle(hThread[0]) == FALSE)
                            {
                                     printf("Error, failed in closing Orders thread handle with errno: %d\n",
GetLastError());
                            }

                            if (CloseHandle(hThread[1]) == FALSE)
                            {
                                     printf("Error, failed in closing NewOrder thread handle with errno: %d\n",
GetLastError());
                            }

                            if (CloseHandle(hThread[2]) == FALSE)
                            {
                                     printf("Error, failed in closing OrderLine thread handle with errno: %d\n",
GetLastError());
                            }
                   }
          }

          printf("Finished loading orders.\n");

    return;
}

//==============================================================================
//
// Function   : OrdersBufInit
//
// Clears shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================================
void OrdersBufInit()
{
          int     i;
          int                j;

          for (i=0;i<orders_per_district;i++)
          {
                   orders_buf[i].o_id = 0;
                   orders_buf[i].o_d_id = 0;
                   orders_buf[i].o_w_id = 0;
                   orders_buf[i].o_c_id = 0;
                   orders_buf[i].o_carrier_id = 0;
                   orders_buf[i].o_ol_cnt = 0;
                   orders_buf[i].o_all_local = 0;

                   for (j=0;j<=14;j++)
                   {
                            orders_buf[i].o_ol[j].ol = 0;
                            orders_buf[i].o_ol[j].ol_i_id = 0;
                            orders_buf[i].o_ol[j].ol_supply_w_id = 0;
                            orders_buf[i].o_ol[j].ol_quantity = 0;
                            orders_buf[i].o_ol[j].ol_amount = 0;
                            strcpy(orders_buf[i].o_ol[j].ol_dist_info,"");
                   }
          }
}

//==============================================================================
//
// Function   : OrdersBufLoad
//
// Fills shared buffer for ORDERS, NEWORDER, and ORDERLINE
//
//==============================================================================
```

```
void OrdersBufLoad(short d_id, long w_id)
{

          int     cust[ORDERS_PER_DISTRICT+1];
          long    o_id;
          long      ol;

          printf("...Loading Order Buffer for: d_id = %d, w_id = %d\n",
                              d_id, w_id);

          GetPermutation(cust, orders_per_district);

          for (o_id=0;o_id<orders_per_district;o_id++)
          {
                    // Generate ORDER and NEW-ORDER data
                    orders_buf[o_id].o_d_id = d_id;
                    orders_buf[o_id].o_w_id = w_id;
                    orders_buf[o_id].o_id = o_id+1;
                    orders_buf[o_id].o_c_id = cust[o_id+1];
                    orders_buf[o_id].o_ol_cnt = (short)RandomNumber(5L, 15L);

                    if (o_id < first_new_order)
                    {
                              orders_buf[o_id].o_carrier_id = (short)RandomNumber(1L, 10L);
                              orders_buf[o_id].o_all_local  = 1;
                    }
                    else
                    {
                              orders_buf[o_id].o_carrier_id = 0;
                              orders_buf[o_id].o_all_local  = 1;
                    }

                    for (ol=0; ol<orders_buf[o_id].o_ol_cnt; ol++)
                    {

                              orders_buf[o_id].o_ol[ol].ol = ol+1;
                              orders_buf[o_id].o_ol[ol].ol_i_id = RandomNumber(1L, max_items);
                              orders_buf[o_id].o_ol[ol].ol_supply_w_id = w_id;
                              orders_buf[o_id].o_ol[ol].ol_quantity = 5;
                              MakeAlphaString(24, 24, OL_DIST_INFO_LEN, &orders_buf[o_id].o_ol[ol].ol_dist_info);

                              // Generate ORDER-LINE data
                              if (o_id < first_new_order)
                              {
                                        orders_buf[o_id].o_ol[ol].ol_amount = 0;
                                        // Added to insure ol_delivery_d set properly during load
                                        FormatDate(&orders_buf[o_id].o_ol[ol].ol_delivery_d);

                              }
                              else
                              {
                                        orders_buf[o_id].o_ol[ol].ol_amount = RandomNumber(1,999999)/100.0;
                                        // Added to insure ol_delivery_d set properly during load

                                        // odbc datetime format
                                        strcpy(orders_buf[o_id].o_ol[ol].ol_delivery_d,"1899-12-31 00:00:00.000");
                              }
                    }
          }
}

//==============================================================================
//
// Function   : LoadOrdersTable
//
//==============================================================================
void LoadOrdersTable(LOADER_TIME_STRUCT *orders_time_start)
{
          int         i;
    long        o_id;
          short     o_d_id;
          long                o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
          char                o_entry_d[O_ENTRY_D_LEN+1];
          RETCODE             rc;
          DBINT               rcint;

          // bind ORDER data
    i = 0;
          rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc1);
          rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_c_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
```

```
                HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_carrier_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_ol_cnt, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_all_local, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);
          rc = bcp_bind(o_hdbc1, (BYTE *) &o_entry_d, 0, O_ENTRY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

    for (i = 0; i < orders_per_district; i++)
    {
                o_id         = orders_buf[i].o_id;
                o_d_id       = orders_buf[i].o_d_id;
                o_w_id       = orders_buf[i].o_w_id;
                o_c_id       = orders_buf[i].o_c_id;
                o_carrier_id = orders_buf[i].o_carrier_id;
                o_ol_cnt     = orders_buf[i].o_ol_cnt;
                o_all_local  = orders_buf[i].o_all_local;

                FormatDate(&o_entry_d);

                // send data to server
                rc = bcp_sendrow(o_hdbc1);
                if (rc != SUCCEED)
                          HandleErrorDBC(o_hdbc1);

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

          if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
          {
                    rcint = bcp_done(o_hdbc1);
                    if (rcint < 0)
                              HandleErrorDBC(o_hdbc1);

                    SQLFreeStmt(o_hstmt1, SQL_DROP);
                    SQLDisconnect(o_hdbc1);
                    SQLFreeConnect(o_hdbc1);

                    // if build index after load...
                    if ((aptr->build_index == 1) && (aptr->index_order == 0))
                              BuildIndex("idxordcl");

                    // build non-clustered index
                    if (aptr->build_index == 1)
                              BuildIndex("idxordnc");
          }
}

//==============================================================================
//
// Function   : LoadNewOrderTable
//
//==============================================================================
void LoadNewOrderTable(LOADER_TIME_STRUCT *new_order_time_start)
{
          long                i;
    long        o_id;
    short       o_d_id;
          long                o_w_id;
          RETCODE             rc;
          DBINT               rcint;

          // Bind NEW-ORDER data
          i = 0;
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);
          rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

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

                    rc = bcp_sendrow(o_hdbc2);
                    if (rc != SUCCEED)
                              HandleErrorDBC(o_hdbc2);

                    new_order_rows_loaded++;
```

```
                    CheckForCommit_Big(o_hdbc2, o_hstmt2, new_order_rows_loaded, "new_order", &new_order_time_start-
>time_start);
          }

          if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
          {
                    rcint = bcp_done(o_hdbc2);
                    if (rcint < 0)
                              HandleErrorDBC(o_hdbc2);

                    SQLFreeStmt(o_hstmt2, SQL_DROP);
                    SQLDisconnect(o_hdbc2);
                    SQLFreeConnect(o_hdbc2);

                    // if build index after load...
                    if ((aptr->build_index == 1) && (aptr->index_order == 0))
                              BuildIndex("idxnodcl");
          }
}

//============================================================================
//
// Function   : LoadOrderLineTable
//
//============================================================================
void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
          long               i;
          long               j;
    long        o_id;
          short     o_d_id;
          long               o_w_id;
          double             ol;
          long      ol_i_id;
          long               ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
          char               ol_delivery_d[OL_DELIVERY_D_LEN+1];
          RETCODE            rc;
          DBINT              rcint;

          // bind ORDER-LINE data
          i = 0;
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_d_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
          rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
          rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_i_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
          rc = bcp_bind(o_hdbc3, (BYTE *) &ol_delivery_d, 0, OL_DELIVERY_D_LEN, NULL, 0, SQLCHARACTER, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_amount, 0, SQL_VARLEN_DATA, NULL, 0, SQLFLT8, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
          rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_quantity, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);
    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, ++i);
          if (rc != SUCCEED)
                    HandleErrorDBC(o_hdbc3);

          for (i = 0; i < orders_per_district; i++)
    {
                    o_id    = orders_buf[i].o_id;
                    o_d_id  = orders_buf[i].o_d_id;
                    o_w_id  = orders_buf[i].o_w_id;

                    for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                    {
                              ol             = orders_buf[i].o_ol[j].ol;
                              ol_i_id        = orders_buf[i].o_ol[j].ol_i_id;
                              ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
                              ol_quantity    = orders_buf[i].o_ol[j].ol_quantity;
                              ol_amount      = orders_buf[i].o_ol[j].ol_amount;
                              strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);

                              strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                              rc = bcp_sendrow(o_hdbc3);
```

```
                                if (rc != SUCCEED)
                                        HandleErrorDBC(o_hdbc3);

                                order_line_rows_loaded++;

                                CheckForCommit_Big(o_hdbc3, o_hstmt3, order_line_rows_loaded, "order_line",
&order_line_time_start->time_start);
                        }
                }

                if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
                {
                        rcint = bcp_done(o_hdbc3);
                        if (rcint < 0)
                                HandleErrorDBC(o_hdbc3);

                        SQLFreeStmt(o_hstmt3, SQL_DROP);
                        SQLDisconnect(o_hdbc3);
                        SQLFreeConnect(o_hdbc3);

                        // if build index after load...
                        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                                BuildIndex("idxodlcl");
                }
}

//==============================================================================
//
// Function   : GetPermutation
//
//==============================================================================
void GetPermutation(int perm[], int n)
{
        int i, r, t;

        for (i=1;i<=n;i++)
                perm[i] = i;

        for (i=1;i<=n;i++)
        {
                r = RandomNumber(i,n);
                t = perm[i];
                perm[i] = perm[r];
                perm[r] = t;
        }
}

//==============================================================================
//
// Function   : CheckForCommit
//
//==============================================================================
void CheckForCommit(HDBC hdbc,
                                                HSTMT hstmt,
                                                long rows_loaded,
                                                char *table_name,
                                        long *time_start)
{

    long  time_end, time_diff;

    if ( !(rows_loaded % aptr->batch) )
    {
                time_end = (TimeNow() / MILLI);
                time_diff = time_end - *time_start;

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

                *time_start = time_end;
    }

    return;
}

//==============================================================================
//
// Function   : CheckForCommit_Big
//
//==============================================================================
void CheckForCommit_Big(HDBC hdbc,
                                                HSTMT hstmt,
                                                double rows_loaded,
                                                char *table_name,
                                        long *time_start)
{

    long  time_end, time_diff;
```

```
    if ( !(fmod(rows_loaded,aptr->batch) ) )
    {
                time_end = (TimeNow() / MILLI);
                time_diff = time_end - *time_start;

                printf("-> Loaded %ld rows into %s in %ld sec - Total = %.0f (%.2f rps)\n",
                          aptr->batch,
                          table_name,
                                    time_diff,
                          rows_loaded,
                                    (float) aptr->batch / (time_diff ? time_diff : 1L));

                *time_start = time_end;
    }

    return;
}

//====================================================================
//
// Function   : OpenConnections
//
//====================================================================
void OpenConnections()
{

          RETCODE         rc;

          char                          szDriverString[300];
          char                          szDriverStringOut[1024];
          SQLSMALLINT                   cbDriverStringOut;


          SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

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

          SQLAllocHandle(SQL_HANDLE_DBC, henv , &i_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &w_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &c_hdbc2);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc1);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc2);
          SQLAllocHandle(SQL_HANDLE_DBC, henv , &o_hdbc3);

          SQLSetConnectAttr(i_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(w_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(c_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(c_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc1, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc2, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );
          SQLSetConnectAttr(o_hdbc3, SQL_COPT_SS_BCP, (void *)SQL_BCP_ON, SQL_IS_INTEGER );

          // Open connections to SQL Server
          // Connection 1
          sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                          aptr->server,
                                                          aptr->user,
                                                          aptr->password,
                                                          aptr->database );

          rc = SQLSetConnectOption (i_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
          if (rc != SUCCEED)
                    HandleErrorDBC(i_hdbc1);

          rc = SQLDriverConnect ( i_hdbc1,
                                                  NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
                                                        (SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
          if ( (rc != SUCCEED) &&
                     (rc != SQL_SUCCESS_WITH_INFO) )
          {
                    HandleErrorDBC(i_hdbc1);
                    printf("TPC-C Loader aborted!\n");
                    exit(9);
          }

          // Connection 2
          sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                          aptr->server,
                                                          aptr->user,
                                                          aptr->password,
                                                          aptr->database );

          rc = SQLSetConnectOption (w_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);

          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);
```

```
rc = SQLDriverConnect ( w_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
                                                        (SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );

if ( (rc != SUCCEED) &&
          (rc != SQL_SUCCESS_WITH_INFO) )
{
         HandleErrorDBC(w_hdbc1);
         printf("TPC-C Loader aborted!\n");
         exit(9);
}

// Connection 3
sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                  aptr->server,
                                                  aptr->user,
                                                  aptr->password,
                                                  aptr->database );

rc = SQLSetConnectOption (c_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
         HandleErrorDBC(c_hdbc1);

rc = SQLDriverConnect ( c_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
                                                        (SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
          (rc != SQL_SUCCESS_WITH_INFO) )
{
         HandleErrorDBC(c_hdbc1);
         printf("TPC-C Loader aborted!\n");
         exit(9);
}

// Connection 4
sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                  aptr->server,
                                                  aptr->user,
                                                  aptr->password,
                                                  aptr->database );

rc = SQLSetConnectOption (c_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
         HandleErrorDBC(c_hdbc2);

rc = SQLDriverConnect ( c_hdbc2,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
                                                        (SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
          (rc != SQL_SUCCESS_WITH_INFO) )
{
         HandleErrorDBC(c_hdbc2);
         printf("TPC-C Loader aborted!\n");
         exit(9);
}

// Connection 5
sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                  aptr->server,
                                                  aptr->user,
                                                  aptr->password,
                                                  aptr->database );

rc = SQLSetConnectOption (o_hdbc1, SQL_PACKET_SIZE, aptr->pack_size);
if (rc != SUCCEED)
         HandleErrorDBC(o_hdbc1);

rc = SQLDriverConnect ( o_hdbc1,
                                                        NULL,
                                                        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,
                                                        (SQLCHAR*)&szDriverStringOut[0],
                                                        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );
if ( (rc != SUCCEED) &&
          (rc != SQL_SUCCESS_WITH_INFO) )
{
```

```
                HandleErrorDBC(o_hdbc1);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 6
        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

        rc = SQLSetConnectOption (o_hdbc2, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc2);

        rc = SQLDriverConnect ( o_hdbc2,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0] ,
                                                                SQL_NTS,
                                                                (SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                  (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(o_hdbc2);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }

        // Connection 7
        sprintf( szDriverString , "DRIVER={SQL Server};SERVER=%s;UID=%s;PWD=%s;DATABASE=%s" ,
                                                        aptr->server,
                                                        aptr->user,
                                                        aptr->password,
                                                        aptr->database );

        rc = SQLSetConnectOption (o_hdbc3, SQL_PACKET_SIZE, aptr->pack_size);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc3);

        rc = SQLDriverConnect ( o_hdbc3,
                                                                NULL,
                                                                (SQLCHAR*)&szDriverString[0] ,
                                                                SQL_NTS,
                                                                (SQLCHAR*)&szDriverStringOut[0],
                                                                sizeof(szDriverStringOut),
                                                                &cbDriverStringOut,
                                                                SQL_DRIVER_NOPROMPT );
        if ( (rc != SUCCEED) &&
                  (rc != SQL_SUCCESS_WITH_INFO) )
        {
                HandleErrorDBC(o_hdbc3);
                printf("TPC-C Loader aborted!\n");
                exit(9);
        }
}

//==============================================================================
//
// Function name: BuildIndex
//
//==============================================================================
void BuildIndex(char          *index_script)
{
        char      cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "osql -S%s -U%s -P%s -e -i%s\\%s.sql > %s%s.log",
                                    aptr->server,
                                    aptr->user,
                                    aptr->password,
                                    aptr->index_script_path,
                                    index_script,
                                    aptr->log_path,
                                    index_script);

        system(cmd);

        printf("Finished index creation:  %s\n",index_script);
}

//==============================================================================
//
// Function name: HandleErrorDBC
//
//==============================================================================
void HandleErrorDBC (SQLHDBC  hdbc1)
{
```

```
          SQLCHAR             SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
          SQLLEN              NativeError;
          SQLSMALLINT i, MsgLen;
          SQLRETURN   rc2;
          char                timebuf[128];
          char                datebuf[128];
          char                err_log_path[256];
          FILE                *fp1;

          i = 1;
          while (( rc2 = SQLGetDiagRec(SQL_HANDLE_DBC , hdbc1, i, SqlState , &NativeError,
                                       Msg, sizeof(Msg) , &MsgLen )) != SQL_NO_DATA )
          {
                    sprintf( szLastError , "%s" ,  Msg );

                    _strtime(timebuf);
                    _strdate(datebuf);

                    printf( "[%s : %s] %s\n==>SQLState: %s\n" , datebuf, timebuf, szLastError, SqlState);

                    strcpy(err_log_path,aptr->log_path);
                    strcat(err_log_path,"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,"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   : 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;
}
//        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])
                  {
                  case '?':     /* Fall throught */
                                    GetArgsLoaderUsage();
                                    break;

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

                  case 'P':
                                    pargs->password = ptr+2;
                                    break;

                  case 'S':
                                    pargs->server = ptr+2;
                                    break;

                  case 'U':
                                    pargs->user = ptr+2;
                                    break;
```

```
                    case 'b':
                                        pargs->batch = atol(ptr+2);
                                        break;

                    case 'W':
                                        pargs->num_warehouses = atol(ptr+2);
                                        break;

                    case 's':
                                        pargs->starting_warehouse = atol(ptr+2);
                                        break;

                    case 't':
                                        {
                                                pargs->tables_all = FALSE;
                                                if (strcmp(ptr+2,"item") == 0)
                                                        pargs->table_item = TRUE;
                                                else if (strcmp(ptr+2,"warehouse") == 0)
                                                        pargs->table_warehouse = TRUE;
                                                else if (strcmp(ptr+2,"customer") == 0)
                                                        pargs->table_customer = TRUE;
                                                else if (strcmp(ptr+2,"orders") == 0)
                                                        pargs->table_orders = TRUE;
                                                else
                                                {
                                                printf("\nUnrecognized command");
                                                GetArgsLoaderUsage();
                                                exit(1);
                                                }

                                                break;
                                        }

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

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

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

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

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

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

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

                            default:
                            GetArgsLoaderUsage();
                            exit(-1);
                            break;
                    }
        }

        /* check for required args */
        if (pargs->num_warehouses == UNDEF )
        {
                    printf("Number of Warehouses is required\n");
                    exit(-2);
        }

        return;
}


//==============================================================================
//
// Function name: GetArgsLoaderUsage
//
//==============================================================================

void GetArgsLoaderUsage()
{

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

```
    printf("TPCCLDR:\n\n");
          printf("Parameter                                                    Default\n");
          printf("-------------------------------------------------------------\n");
    printf("-W Number of Warehouses to Load                        Required \n");
    printf("-S Server                                              %s\n", SERVER);
    printf("-U Username                                            %s\n", USER);
    printf("-P Password                                            %s\n", PASSWORD);
    printf("-D Database                                            %s\n", DATABASE);
          printf("-b Batch Size                                          %ld\n", (long) BATCH);
          printf("-p TDS packet size                                     %ld\n", (long) DEFLDPACKSIZE);
          printf("-L Loader BCP Log Path                                 %s\n", LOADER_LOG_PATH);
          printf("-f Loader Results Output Filename                      %s\n", LOADER_RES_FILE);
          printf("-s Starting Warehouse                                  %ld\n", (long) DEF_STARTING_WAREHOUSE);
          printf("-i Build Option (data = 0, data and index = 1)         %ld\n", (long) BUILD_INDEX);
          printf("-o Cluster Index Build Order (before = 1, after = 0)  %ld\n", (long) INDEX_ORDER);
          printf("-c Build Scaled Database (normal = 0, tiny = 1)        %ld\n", (long) SCALE_DOWN);
          printf("-d Index Script Path                                   %s\n", INDEX_SCRIPT_PATH);
          printf("-t Table to Load                                       all tables \n");
    printf("   [item|warehouse|customer|orders]\n");
    printf("   Notes: \n");
    printf("     - the '-t' parameter may be included multiple times to \n");
    printf("       specify multiple tables to be loaded \n");
    printf("     - 'item' loads ITEM table \n");
    printf("     - 'warehouse' loads WAREHOUSE, DISTRICT, and STOCK tables \n");
    printf("     - 'customer' loads CUSTOMER and HISTORY tables \n");
    printf("     - 'orders' load NEW-ORDER, ORDERS, ORDER-LINE tables \n");

          printf("\nNote:  Command line switches are case sensitive.\n");

    exit(0);
}


//        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("[%ld]DBG: Entering irand()...\n", (int) GetCurrentThreadId());
#endif

    s = Seed;
    hi = s / Q;
    lo = s % Q;

    test = A * lo - R * hi;
    if ( test > 0 )
                    Seed = test;
    else
                    Seed = test + M;

    return( Seed );
}


/*****************************************************************************
*                                                                           *
* drand - returns a double pseudo random number between 0.0 and 1.0.        *
*       See irand.                                                          *
*****************************************************************************/
double drand()
{

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

    return( (double)irand() / 2147483647.0);
}


//===========================================================================
// Function   : RandomNumber
//
// Description:
//===========================================================================
long RandomNumber(long lower, long upper)
{
    long rand_num;

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

          if ( upper == lower )         /* pgd 08-13-96 perf enhancement */
                    return lower;

          upper++;

    if ( upper <= lower )
                    rand_num = upper;
    else
                    rand_num = lower + irand() % (upper - lower); /* pgd 08-13-96 perf enhancement */


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                        (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}


#if 0

//Orginal code pgd 08/13/96

long RandomNumber(long lower,
                                        long upper)
```

```
{
    long rand_num;

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

          upper++;

    if ((upper <= lower))
                    rand_num = upper;
    else
                    rand_num = lower + irand() % ((upper > lower) ? upper - lower : upper);


#ifdef DEBUG
    printf("[%ld]DBG: RandomNumber between %ld & %ld ==> %ld\n",
                                        (int) GetCurrentThreadId(), lower, upper, rand_num);
#endif


    return rand_num;
}
#endif

//==============================================================================
// Function   : NURand
//
// Description:
//==============================================================================
long NURand(int iConst,
              long x,
              long y,
              long C)
{
    long rand_num;

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

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

#ifdef DEBUG
    printf("[%ld]DBG: NURand: num = %d\n", (int) GetCurrentThreadId(), rand_num);
#endif

    return rand_num;
}


//        File:               STRINGS.C
//                                    Microsoft TPC-C Kit Ver. 4.51
//                                    Copyright Microsoft, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003
//        Purpose:  Source file for database loader string functions


// Includes
#include "tpcc.h"
#include <string.h>
#include <ctype.h>

//==============================================================================
//
// Function name: MakeAddress
//
//==============================================================================

void MakeAddress(char *street_1,
                                     char *street_2,
                                     char *city,
                                     char *state,
                                     char *zip)
{

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

    MakeAlphaString (10, 20, ADDRESS_LEN, street_1);
    MakeAlphaString (10, 20, ADDRESS_LEN, street_2);
    MakeAlphaString (10, 20, ADDRESS_LEN, city);
    MakeAlphaString ( 2,  2, STATE_LEN, state);
    MakeZipNumberString( 9,  9, ZIP_LEN, zip);

#ifdef DEBUG
    printf("[%ld]DBG: MakeAddress: street_1: %s, street_2: %s, city: %s, state: %s, zip: %s\n",
                            (int) GetCurrentThreadId(), street_1, street_2, city, state, zip);
#endif

    return;

}
```

```
//====================================================================
//
// Function name: LastName
//
//====================================================================

void LastName(int  num,
                        char *name)
{
    static char *n[] =
    {
                "BAR" , "OUGHT", "ABLE" , "PRI"  , "PRES",
                "ESE" , "ANTI" , "CALLY", "ATION", "EING"
    };

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

    if ((num >= 0) && (num < 1000))
    {
                strcpy(name, n[(num/100)%10]);
                strcat(name, n[(num/10)%10]);
                strcat(name, n[(num/1)%10]);

                if (strlen(name) < LAST_NAME_LEN)
                {
                        PaddString(LAST_NAME_LEN, name);
                }

    }
    else
    {
                printf("\nError in LastName()... num <%ld> out of range (0,999)\n", num);
                exit(-1);
    }


#ifdef DEBUG
    printf("[%ld]DBG: LastName: num = [%d] ==> [%d][%d][%d]\n",
                        (int) GetCurrentThreadId(), num, num/100, (num/10)%10, num%10);
        printf("[%ld]DBG: LastName: String = %s\n", (int) GetCurrentThreadId(), name);
#endif

    return;

}


//====================================================================
//
// Function name: MakeAlphaString
//
//====================================================================

//philipdu 08/13/96 Changed MakeAlphaString to use A-Z, a-z, and 0-9 in
//accordance with spec see below:
//The spec says:
//4.3.2.2 The notation random a-string [x .. y]
//(respectively, n-string [x .. y]) represents a string of random alphanumeric
//(respectively, numeric) characters of a random length of minimum x, maximum y,
//and mean (y+x)/2. Alphanumerics are A..Z, a..z, and 0..9.  The only other
//requirement is that the character set used "must be able to represent a minimum
//of 128 different characters".  We are using 8-bit chars, so this is a non issue.
//It is completely unreasonable to stuff non-printing chars into the text fields.
//-CLevine 08/13/96

int MakeAlphaString( int  x, int  y, int  z, char *str)
{
        int                 len;
        int                 i;
    char    cc = 'a';
        static    char chArray[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
        static    int       chArrayMax = 61;

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

        len= RandomNumber(x, y);

        for (i=0; i<len; i++)
                str[i] = chArray[RandomNumber(0,chArrayMax)];
        str[len] = 0;

    return len;
}

int MakeAlphaStringPadded( int  minLen, int  maxLen, int  padLen, char *str)
{
        int                 len;
```

```
          int                 i;
    char    cc = 'a';
          static    char chArray[] = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
          static    int       chArrayMax = 61;

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

          len= RandomNumber(minLen, maxLen);

          for (i=0; i<len; i++)
                    str[i] = chArray[RandomNumber(0,chArrayMax)];
          if (len < padLen)
                    memset(str+len, ' ', padLen - len);
          str[padLen] = 0;
    return padLen;
}



//====================================================================
//
// Function name: MakeOriginalAlphaString
//
//====================================================================

int MakeOriginalAlphaString(int  x,
                                                                int  y,
                                                                     int  z,
                                                                char *str,
                                                           int  percent)
{
    int             len;
    int             val;
    int             start;

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

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

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

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

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                    start = RandomNumber(0, len - 8);
                    strncpy(str + start, "ORIGINAL", 8);
    }


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

    return len;
}


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

          //MakeNumberString is always called MakeZipNumberString(16, 16, 16, string)

          memset(str, '0', 16);
          itoa(RandomNumber(0, 99999999), tmp, 10);
          memcpy(str, tmp, strlen(tmp));

          itoa(RandomNumber(0, 99999999), tmp, 10);
          memcpy(str+8, tmp, strlen(tmp));
```

```
          str[16] = 0;

    return 16;
}


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

          //MakeZipNumberString is always called MakeZipNumberString(9, 9, 9, string)

          strcpy(str, "000011111");

          itoa(RandomNumber(0, 9999), tmp, 10);
          memcpy(str, tmp, strlen(tmp));

    return 9;
}


//==============================================================================
//
// Function name: InitString
//
//==============================================================================
void InitString(char *str, int len)
{
#ifdef DEBUG
    printf("[%ld]DBG: Entering InitString()\n", (int) GetCurrentThreadId());
#endif

          memset(str, ' ', len);
    str[len] = 0;
}


//==============================================================================
// Function name: InitAddress
//
// Description:
//
//==============================================================================

void InitAddress(char *street_1, char *street_2, char *city, char *state, char *zip)
{
          memset(street_1, ' ', ADDRESS_LEN+1);
          memset(street_2, ' ', ADDRESS_LEN+1);
          memset(city, ' ', ADDRESS_LEN+1);

    street_1[ADDRESS_LEN+1] = 0;
    street_2[ADDRESS_LEN+1] = 0;
    city[ADDRESS_LEN+1]     = 0;

          memset(state, ' ', STATE_LEN+1);
    state[STATE_LEN+1] = 0;

          memset(zip, ' ', ZIP_LEN+1);
    zip[ZIP_LEN+1] = 0;
}


//==============================================================================
//
// Function name: PaddString
//
//==============================================================================

void PaddString(int max, char *name)
{
          int                len;

          len = strlen(name);
          if ( len < max )
                    memset(name+len, ' ', max - len);
          name[max] = 0;

          return;
}
//        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;
```

Appendix C - Tunable Parameters

Server Configuration Parameters

Microsoft Windows 2003 Server Parameters

Microsoft Windows 2003 Server Configuration

The following services were disabled on the server:

* Alerter
* Automatic Updates
* Computer Browser
* Cryptographic Services
* DHCP Client
* Distributed File System
* Distributed Link Tracking Client
* DNS Client
* Global Array Manager Server
* Help and Support
* IPSEC Policy Agent
* License Logging Service
* Messenger
* MSSQLserver
* Microsoft Search
* Print Spooler
* Process Control Service
* Remote Registry Service
* Removable Storage
* Run as Service
* System Event Notification
* SSDP Discovery service
* Task Scheduler
* Wireless configuration

Microsoft SQL Server 2005 Startup Parameters

Microsoft SQL Server was started with the following command line options

sqlservr -c -x -T3502 –T8011 –T8012 –T8018 –T8019 –T661 –T836

where

-c	Start SQL Server independently of the Microsoft Windows NT Service Control Manager.
-x	Disable the keeping of CPU time and cache-hit ratio statistics.
-T3502	Prints a message to the log at the beginning and end of each checkpoint.
-T661	Disable ghost writer
-T8011	Disable diagnostics for resource monitor
-T8012	Disable ring buffer for scheduler
-T8018	Disable exceptions ring buffer
-T8019	Disable stack collection for exception ring buffer
-T836	Force max server memory
-T834	Force buffer pool to use large pages

Microsoft SQL Server 2005 Configuration Parameters

```
name                                 minimum     maximum     config_value run_value
------------------------------------ ----------- ----------- ------------
       -----------
        -----------
Ad Hoc Distributed Queries                     0           1            0
                  0
affinity I/O mask                     -2147483648  2147483647            0
                  0
affinity mask                         -2147483648  2147483647            3
                  3
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            1
                  1
blocked process threshold                      0       86400            0
                  0
c2 audit mode                                  0           1            0
                  0
clr enabled                                    0           1            0
                  0
cost threshold for parallelism                 0       32767            0
                  0
cross db ownership chaining                    0           1            0
                  0
cursor threshold                              -1  2147483647           -1
                 -1
Database Mail XPs                              0           1            0
                  0
default full-text language                     0  2147483647         1033
               1033
default language                               0        9999            0
                  0
default trace enabled                          0           1            1
                  1
disallow results from triggers                 0           1            0
                  0
fill factor (%)                                0         100            0
                  0
ft crawl bandwidth (max)                       0       32767          100
                100
ft crawl bandwidth (min)                       0       32767            0
                  0
ft notify bandwidth (max)                      0       32767          100
                100
ft notify bandwidth (min)                      0       32767            0
                  0
in-doubt xact resolution                       0           2            0
                  0
index create memory (KB)                     704  2147483647            0
                  0
lightweight pooling                            0           1            1
                  1
locks                                       5000  2147483647            0
                  0
max degree of parallelism                      0          64            0
                  0
max full-text crawl range                      0         256            4
```

```
                4
max server memory (MB)                 16  2147483647   2147483647
      2147483647
max text repl size (B)                  0  2147483647        65536
           65536
max worker threads                    128       32767          720
             720
media retention                         0         365            0
               0
min memory per query (KB)             512  2147483647         1024
            1024
min server memory (MB)                  0  2147483647            0
               0
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           20
              20
remote proc trans                       0           1            0
               0
remote query timeout (s)                0  2147483647          600
             600
Replication XPs                         0           1            0
               0
scan for startup procs                  0           1            0
               0
server trigger recursion                0           1            1
               1
set working set size                    0           1            0
               0
show advanced options                   0           1            1
               1
SMO and DMO XPs                         0           1            1
               1
SQL Mail XPs                            0           1            0
               0
transform noise words                   0           1            0
               0
two digit year cutoff                1753        9999         2049
            2049
user connections                        0       32767            0
               0
user options                            0       32767            0
               0
Web Assistant Procedures                0           1            0
               0
xp_cmdshell                             0           1            0
```

Appendix C – Tunable Parameters

System Information report written at: 02/26/07 11:22:39
System Name: PE2900
[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003 Standard x64 Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	PE2900
System Manufacturer	Dell Inc.
System Model	PowerEdge 2900
System Type	x64-based PC
Processor	EM64T Family 6 Model 15 Stepping 7 GenuineIntel ~2328 Mhz
Processor	EM64T Family 6 Model 15 Stepping 7 GenuineIntel ~2328 Mhz
Processor	EM64T Family 6 Model 15 Stepping 7 GenuineIntel ~2328 Mhz
Processor	EM64T Family 6 Model 15 Stepping 7 GenuineIntel ~2328 Mhz
BIOS Version/Date	Dell Inc. 1.1.7, 9/28/2006
SMBIOS Version	2.4
Windows Directory	C:\WINDOWS
System Directory	C:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume6
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)"
User Name	Not Available
Time Zone	Central Standard Time
Total Physical Memory	24,574.98 MB
Available Physical Memory	22.97 GB
Total Virtual Memory	49.66 GB
Available Virtual Memory	49.24 GB
Page File Space	26.41 GB
Page File	C:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
I/O Port 0x00000000-0x00000CF7	PCI bus
I/O Port 0x00000000-0x00000CF7	Direct memory access controller
IRQ 20	Standard Universal PCI to USB Host Controller
IRQ 20	Standard Universal PCI to USB Host Controller
Memory Address 0xF4000000-0xF7FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address 0xF4000000-0xF7FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address 0xF4000000-0xF7FFFFFF	PCI standard PCI-to-PCI bridge
Memory Address 0xF4000000-0xF7FFFFFF	Broadcom BCM5708C NetXtreme II GigE
IRQ 21	Standard Universal PCI to USB Host Controller
IRQ 21	Standard Universal PCI to USB Host Controller
IRQ 21	Standard Enhanced PCI to USB Host Controller

Memory Address 0xD8000000-0xD80FFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xD8000000-0xD80FFFFF PCI standard PCI-to-PCI bridge

Memory Address 0xD8100000-0xD81FFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xD8100000-0xD81FFFFF PCI standard PCI-to-PCI bridge

Memory Address 0xD8200000-0xD82FFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xD8200000-0xD82FFFFF PCI standard PCI-to-PCI bridge

IRQ 16 PCI standard host CPU bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 Broadcom BCM5708C NetXtreme II GigE
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 PCI standard PCI-to-PCI bridge
IRQ 16 Broadcom BCM5708C NetXtreme II GigE

Memory Address 0xD0000000-0xFDFFFFFF PCI bus
Memory Address 0xD0000000-0xFDFFFFFF Standard VGA Graphics Adapter

IRQ 18 DELL PERC 5/E Adapter RAID Controller
IRQ 18 DELL PERC 5/E Adapter RAID Controller
IRQ 18 DELL PERC 5/E Adapter RAID Controller

Memory Address 0xA0000-0xBFFFF PCI bus
Memory Address 0xA0000-0xBFFFF Standard VGA Graphics Adapter

Memory Address 0xF8000000-0xFBFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF8000000-0xFBFFFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xF8000000-0xFBFFFFFF Broadcom BCM5708C NetXtreme II GigE

Memory Address 0xD8300000-0xD83FFFFF PCI standard PCI-to-PCI bridge
Memory Address 0xD8300000-0xD83FFFFF PCI standard PCI-to-PCI bridge

[DMA]

Resource Device Status
Channel 4 Direct memory access controller OK
Channel 2 Standard floppy disk controller 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
0x0000DCE0-0x0000DCFF   Standard Universal PCI to USB Host Controller   OK
0x0000DCC0-0x0000DCDF   Standard Universal PCI to USB Host Controller   OK
0x0000DCA0-0x0000DCBF   Standard Universal PCI to USB Host Controller   OK
0x0000DC80-0x0000DC9F   Standard Universal PCI to USB Host Controller   OK
0x0000EC00-0x0000ECFF   Standard VGA Graphics Adapter          OK
0x000003B0-0x000003BB   Standard VGA Graphics Adapter          OK
0x000003C0-0x000003DF   Standard VGA Graphics Adapter          OK
0x00000080-0x0000009F   Direct memory access controller OK
0x000000C0-0x000000DF   Direct memory access controller OK
0x000000F0-0x000000FF   Numeric data processor OK
0x00000020-0x0000003F   Programmable interrupt controller      OK
0x000000A0-0x000000BF   Programmable interrupt controller      OK
0x000004D0-0x000004D1   Programmable interrupt controller      OK
0x00000061-0x00000061   System board  OK
0x00000070-0x0000007F   System CMOS/real time clock   OK
0x00000040-0x0000005F   System timer  OK
0x000003F0-0x000003F5   Standard floppy disk controller   OK
0x000003F7-0x000003F7   Standard floppy disk controller   OK
0x000003F8-0x000003FF   Communications Port (COM1)   OK
0x00000800-0x0000087F   System board  OK
0x00000880-0x000008BF   System board  OK
0x000008C0-0x000008DF   System board  OK
0x000008E0-0x000008E3   System board  OK
0x00000C00-0x00000C7F   System board  OK
0x00000CA0-0x00000CA7   System board  OK
0x00000CA9-0x00000CAB   System board  OK
0x00000CAD-0x00000CAF   System board  OK
0x00000060-0x00000060   System board  OK
0x00000064-0x00000064   System board  OK
0x00000CA8-0x00000CA8   System board  OK
0x00000CAC-0x00000CAC   System board  OK
0x0000FC00-0x0000FC0F   Standard Dual Channel PCI IDE Controller   OK
0x000001F0-0x000001F7   Primary IDE Channel   OK
0x000003F6-0x000003F6   Primary IDE Channel   OK
0x00000170-0x00000177   Secondary IDE Channel OK
0x00000376-0x00000376   Secondary IDE Channel OK

[IRQs]

Resource        Device  Status
IRQ 9   Microsoft ACPI-Compliant System         OK
IRQ 16  PCI standard host CPU bridge    OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  Broadcom BCM5708C NetXtreme II GigE         OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  PCI standard PCI-to-PCI bridge  OK
IRQ 16  Broadcom BCM5708C NetXtreme II GigE         OK
IRQ 18  DELL PERC 5/E Adapter RAID Controller       OK
```

IRQ 18 DELL PERC 5/E Adapter RAID Controller OK
IRQ 18 DELL PERC 5/E Adapter RAID Controller OK
IRQ 142 DELL PERC 5/i Integrated RAID Controller OK
IRQ 21 Standard Universal PCI to USB Host Controller OK
IRQ 21 Standard Universal PCI to USB Host Controller OK
IRQ 21 Standard Enhanced PCI to USB Host Controller OK
IRQ 20 Standard Universal PCI to USB Host Controller OK
IRQ 20 Standard Universal PCI to USB Host Controller OK
IRQ 13 Numeric data processor OK
IRQ 8 System CMOS/real time clock OK
IRQ 0 System timer OK
IRQ 6 Standard floppy disk controller OK
IRQ 4 Communications Port (COM1) OK
IRQ 14 Primary IDE Channel OK

[Memory]

Resource Device Status
0xA0000-0xBFFFF PCI bus OK
0xA0000-0xBFFFF Standard VGA Graphics Adapter OK
0xD0000000-0xFDFFFFFF PCI bus OK
0xD0000000-0xFDFFFFFF Standard VGA Graphics Adapter OK
0xF2000000-0xF7FFFFFF PCI standard PCI-to-PCI bridge OK
0xF4000000-0xF7FFFFFF PCI standard PCI-to-PCI bridge OK
0xF4000000-0xF7FFFFFF PCI standard PCI-to-PCI bridge OK
0xF4000000-0xF7FFFFFF PCI standard PCI-to-PCI bridge OK
0xF4000000-0xF7FFFFFF Broadcom BCM5708C NetXtreme II GigE OK
0xFC900000-0xFCBFFFFF PCI standard PCI-to-PCI bridge OK
0xD8200000-0xD82FFFFF PCI standard PCI-to-PCI bridge OK
0xD8200000-0xD82FFFFF PCI standard PCI-to-PCI bridge OK
0xFCA00000-0xFCBFFFFF PCI standard PCI-to-PCI bridge OK
0xD82F0000-0xD82FFFFF DELL PERC 5/E Adapter RAID Controller OK
0xFCAE0000-0xFCAFFFFF DELL PERC 5/E Adapter RAID Controller OK
0xFC600000-0xFC8FFFFF PCI standard PCI-to-PCI bridge OK
0xD8100000-0xD81FFFFF PCI standard PCI-to-PCI bridge OK
0xD8100000-0xD81FFFFF PCI standard PCI-to-PCI bridge OK
0xFC700000-0xFC8FFFFF PCI standard PCI-to-PCI bridge OK
0xD81F0000-0xD81FFFFF DELL PERC 5/E Adapter RAID Controller OK
0xFC7E0000-0xFC7FFFFF DELL PERC 5/E Adapter RAID Controller OK
0xFCC00000-0xFCEFFFFF PCI standard PCI-to-PCI bridge OK
0xD8300000-0xD83FFFFF PCI standard PCI-to-PCI bridge OK
0xD8300000-0xD83FFFFF PCI standard PCI-to-PCI bridge OK
0xFCD00000-0xFCEFFFFF PCI standard PCI-to-PCI bridge OK
0xD83F0000-0xD83FFFFF DELL PERC 5/i Integrated RAID Controller OK
0xFCDE0000-0xFCDFFFFF DELL PERC 5/i Integrated RAID Controller OK
0xFC300000-0xFC5FFFFF PCI standard PCI-to-PCI bridge OK
0xD8000000-0xD80FFFFF PCI standard PCI-to-PCI bridge OK
0xD8000000-0xD80FFFFF PCI standard PCI-to-PCI bridge OK
0xFC400000-0xFC5FFFFF PCI standard PCI-to-PCI bridge OK
0xD80F0000-0xD80FFFFF DELL PERC 5/E Adapter RAID Controller OK
0xFC4E0000-0xFC4FFFFF DELL PERC 5/E Adapter RAID Controller OK
0xF8000000-0xFBFFFFFF PCI standard PCI-to-PCI bridge OK
0xF8000000-0xFBFFFFFF PCI standard PCI-to-PCI bridge OK
0xF8000000-0xFBFFFFFF Broadcom BCM5708C NetXtreme II GigE OK
0xFCF00000-0xFCF003FF Standard Enhanced PCI to USB Host Controller OK

0xFC1F0000-0xFC1FFFFF Standard VGA Graphics Adapter OK
0xE0000000-0xEFFFFFFF Motherboard resources OK
0xFED00000-0xFED003FF High precision event timer OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC Manufacturer Description Status File Version Size Creation Date
c:\windows\system32\tssoft32.acm DSP GROUP, INC. OK
C:\WINDOWS\system32\TSSOFT32.ACM 1.01 13.50 KB (13,824 bytes) 3/25/2005 6:00 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 6:00 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 6:00 AM
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 6:00 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 6:00 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) 52.50 KB (53,760 bytes)3/24/2005 11:19 AM
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 6:00 AM
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 6:00 AM
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 11:21 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 11:34 AM

[CD-ROM]

Item Value
Drive D:
Description CD-ROM Drive

Media Loaded Yes
Media Type CD-ROM
Name LITE-ON CD-ROM LTN-4891S
Manufacturer (Standard CD-ROM drives)
Status OK
Transfer Rate 2153.85 kbytes/sec
SCSI Target ID 0
PNP Device ID IDE\CDROMLITE-ON_CD-ROM_LTN-4891S_____NDS3____\5&41A3CB2&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 6:00 AM)

[Sound Device]

Item Value

[Display]

Item Value
Name Standard VGA Graphics Adapter
PNP Device ID
PCI\VEN_1002&DEV_515E&SUBSYS_01B11028&REV_02\4&2014205D&0&68F0
Adapter Type Not Available
Adapter Description (Standard display types)
Adapter RAM Not Available
Installed Drivers Not Available
Driver Version Not Available
INF File 5.2.3790.1830 (display.inf section)
Color Planes vga
Color Table Entries Not Available
Resolution Not Available
Bits/Pixel Not Available
Memory Address 0xD0000000-0xFDFFFFFF
I/O Port 0x0000EC00-0x0000ECFF
Memory Address 0xFC1F0000-0xFC1FFFFF
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver c:\windows\system32\drivers\vgapnp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 33.00 KB (33,792 bytes), 2/23/2007 6:41 AM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description USB Human Interface Device
Name Enhanced (101- or 102-key)
Layout 00000409

PNP Device ID USB\VID_0557&PID_2221&MI_00\6&1D334AC&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 6:00 AM)

[Pointing Device]

Item Value
Hardware Type USB Human Interface Device
Number of Buttons 3
Status OK
PNP Device ID USB\VID_0557&PID_2221&MI_01\6&1D334AC&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 6:00 AM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 2/25/2007 11:33 AM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPORT\0000
Last Reset 2/25/2007 11:33 AM
Index 2
Service Name Rasl2tp
IP Address Not Available

IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 132.00 KB (135,168 bytes), 3/25/2005 6:00 AM)

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPORT\0000
Last Reset 2/25/2007 11:33 AM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver c:\windows\system32\drivers\raspptp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 117.50 KB (120,320 bytes), 3/25/2005 6:00 AM)

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPORT\0000
Last Reset 2/25/2007 11:33 AM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver c:\windows\system32\drivers\raspppoe.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 67.50 KB (69,120 bytes), 3/25/2005 6:00 AM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTIMINIPORT\0000
Last Reset 2/25/2007 11:33 AM
Index 5

Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\raspti.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 30.50 KB (31,232 bytes), 3/25/2005 6:00 AM)

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 2/25/2007 11:33 AM
Index 6
Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver c:\windows\system32\drivers\ndiswan.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 157.50 KB (161,280 bytes), 3/25/2005 6:00 AM)

Name [00000007] 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
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_01B11028&REV_12\6&2F68317E&0&20050500
Last Reset 2/25/2007 11:33 AM
Index 7
Service Name l2nd
IP Address 192.1.1.51
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 00:13:72:64:D1:48
Driver c:\windows\system32\drivers\bxnd52a.sys (2.6.14.0 built by: WinDDK, 78.00 KB (79,872 bytes), 4/3/2006 2:53 PM)

Name [00000008] 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
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_01B11028&REV_12\8&126A2D63&0&2005
0900
Last Reset 2/25/2007 11:33 AM
Index 8
Service Name l2nd
IP Address 192.1.1.50
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 00:13:72:64:D1:4A
Driver c:\windows\system32\drivers\bxnd52a.sys (2.6.14.0 built by: WinDDK, 78.00 KB (79,872 bytes), 4/3/2006 2:53 PM)

[Protocol]

Item Value
Name MSAFD Tcpip [TCP/IP]
Connectionless Service No
Guarantees Delivery Yes
Guarantees SequencingYes
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 SequencingNo
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 SequencingNo
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 SequencingYes
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 0
XOffXMit Threshold 0
XOn Character 0
XOnXMit Threshold 0
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 6:00 AM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value
Drive A:
Description 3 1/2 Inch Floppy Drive

Drive B:
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 C:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 10.00 GB (10,733,957,120 bytes)
Free Space 4.73 GB (5,075,996,672 bytes)
Volume Name
Volume Serial Number FC1B67A7

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 Removable Disk

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive O:
Description 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 P:
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 R:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 822.27 GB (882,901,516,288 bytes)
Free Space 616.06 GB (661,491,875,840 bytes)
Volume Name
Volume Serial Number C8593D9A

Drive T:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 853.84 GB (916,806,107,136 bytes)
Free Space 623.22 GB (669,181,739,008 bytes)
Volume Name
Volume Serial Number D06D8106

Drive V:
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 W:
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

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)
Model DELL PERC 5/E Adapter SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 1
SCSI Logical Unit 0
SCSI Port 3

SCSI Target ID 0
Sectors/Track 63
Size 1,001.24 GB (1,075,076,997,120 bytes)
Total Cylinders 130,704
Total Sectors 2,099,759,760
Total Tracks 33,329,520
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 119.99 GB (128,840,753,664 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #1, Partition #1
Partition Size 58.98 GB (63,334,656,000 bytes)
Partition Starting Offset 128,840,785,920 bytes
Partition Disk #1, Partition #2
Partition Size 822.27 GB (882,901,555,200 bytes)
Partition Starting Offset 192,175,441,920 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model DELL PERC 5/E Adapter SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 1
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 1,001.24 GB (1,075,076,997,120 bytes)
Total Cylinders 130,704
Total Sectors 2,099,759,760
Total Tracks 33,329,520
Tracks/Cylinder 255
Partition Disk #2, Partition #0
Partition Size 98.82 GB (106,106,079,744 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #2, Partition #1
Partition Size 48.58 GB (52,164,725,760 bytes)
Partition Starting Offset 106,106,112,000 bytes
Partition Disk #2, Partition #2
Partition Size 853.84 GB (916,806,159,360 bytes)
Partition Starting Offset 158,270,837,760 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model DELL PERC 5/E Adapter 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 2
SCSI Target ID 0

Sectors/Track 63
Size 1,001.24 GB (1,075,076,997,120 bytes)
Total Cylinders 130,704
Total Sectors 2,099,759,760
Total Tracks 33,329,520
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 98.82 GB (106,106,079,744 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #0, Partition #1
Partition Size 48.58 GB (52,164,725,760 bytes)
Partition Starting Offset 106,106,112,000 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model DELL PERC 5/i 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 5
SCSI Target ID 0
Sectors/Track 63
Size 544.49 GB (584,644,677,120 bytes)
Total Cylinders 71,079
Total Sectors 1,141,884,135
Total Tracks 18,125,145
Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 10.00 GB (10,733,958,144 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #3, Partition #1
Partition Size 534.50 GB (573,910,686,720 bytes)
Partition Starting Offset 10,733,990,400 bytes

[SCSI]

Item Value
Name DELL PERC 5/E Adapter RAID Controller
Manufacturer DELL
Status OK
PNP Device ID
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&1E758DE3&0&700018
Memory Address 0xD82F0000-0xD82FFFFF
Memory Address 0xFCAE0000-0xFCAFFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\percsas.sys (1.20.0.64 built by: WinDDK, 26.50 KB (27,136 bytes), 2/23/2007 6:31 AM)

Name DELL PERC 5/E Adapter RAID Controller
Manufacturer DELL
Status OK

PNP Device ID
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&20524F73&0&700020
Memory Address 0xD81F0000-0xD81FFFFF
Memory Address 0xFC7E0000-0xFC7FFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\percsas.sys (1.20.0.64 built by: WinDDK, 26.50 KB (27,136 bytes), 2/23/2007 6:31 AM)

Name DELL PERC 5/i Integrated RAID Controller
Manufacturer DELL
Status OK
PNP Device ID
PCI\VEN_1028&DEV_0015&SUBSYS_1F031028&REV_00\5&22FD9970&0&700028
Memory Address 0xD83F0000-0xD83FFFFF
Memory Address 0xFCDE0000-0xFCDFFFFF
IRQ Channel IRQ 142
Driver c:\windows\system32\drivers\percsas.sys (1.20.0.64 built by: WinDDK, 26.50 KB (27,136 bytes), 2/23/2007 6:31 AM)

Name DELL PERC 5/E Adapter RAID Controller
Manufacturer DELL
Status OK
PNP Device ID
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&376DDE58&0&700030
Memory Address 0xD80F0000-0xD80FFFFF
Memory Address 0xFC4E0000-0xFC4FFFFF
IRQ Channel IRQ 18
Driver c:\windows\system32\drivers\percsas.sys (1.20.0.64 built by: WinDDK, 26.50 KB (27,136 bytes), 2/23/2007 6:31 AM)

[IDE]

Item Value
Name Standard Dual Channel PCI IDE Controller
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\VEN_8086&DEV_269E&SUBSYS_01B11028&REV_09\3&61AAA01&0&F9
I/O Port 0x0000FC00-0x0000FC0F
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 6:00 AM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&1D8A9C03&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), 145.00 KB (148,480 bytes), 3/25/2005 6:00 AM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&1D8A9C03&0&1

I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
Driver c:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 6:00 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_2688&SUBSYS_01B11028&REV_09\3&61AAA01&0&E8
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_2689&SUBSYS_01B11028&REV_09\3&61AAA01&0&E9
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_268A&SUBSYS_01B11028&REV_09\3&61AAA01&0&EA
Standard Universal PCI to USB Host Controller
PCI\VEN_8086&DEV_268B&SUBSYS_01B11028&REV_09\3&61AAA01&0&EB
Standard Enhanced PCI to USB Host Controller
PCI\VEN_8086&DEV_268C&SUBSYS_01B11028&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 No No
acpi Microsoft ACPI Driver c:\windows\system32\drivers\acpi.sys Kernel Driver Yes
Boot Running OK Normal No Yes
acpiec ACPIEC c:\windows\system32\drivers\acpiec.sys Kernel Driver No
Disabled Stopped OK Normal No No
adpu160m adpu160m Not Available Kernel Driver No Disabled
Stopped OK Normal No No
adpu320 adpu320 Not Available Kernel Driver No Disabled
Stopped OK Normal No No
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
asyncmac RAS Asynchronous Media Driver
c:\windows\system32\drivers\asyncmac.sys Kernel Driver No Manual
Stopped OK Normal No No
atapi Standard IDE/ESDI Hard Disk Controller c:\windows\system32\drivers\atapi.sys Kernel
Driver Yes Boot Running OK Normal No Yes
atdisk Atdisk Not Available Kernel Driver No Disabled Stopped OK
Ignore No No
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
beep Beep c:\windows\system32\drivers\beep.sys Kernel Driver Yes System
Running OK Normal No Yes
blfp Broadcom Advanced Server Program Driver
c:\windows\system32\drivers\basamd64.sys Kernel Driver No Manual
Stopped OK Normal No No
cdac15ba CdaC15BA c:\windows\system32\drivers\cdac15ba.sys Kernel Driver
Yes Auto Running OK Normal No Yes
cdad10ba CdaD10BA c:\windows\system32\drivers\cdad10ba.sys Kernel Driver
Yes Auto Running OK Normal No Yes
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
cpqcissm cpqcissm Not Available Kernel Driver No Disabled
Stopped OK Normal No No
crcdisk CRC Disk Filter Driver c:\windows\system32\drivers\crcdisk.sys Kernel Driver Yes
Boot Running OK Normal No Yes
dfsdriver DfsDriver c:\windows\system32\drivers\dfs.sys File System Driver
Yes Boot Running OK Normal No Yes
disk Disk Driver c:\windows\system32\drivers\disk.sys Kernel Driver Yes Boot
Running OK Normal No Yes
dmboot dmboot c:\windows\system32\drivers\dmboot.sys Kernel Driver No
Disabled Stopped OK Normal No No
dmio Logical Disk Manager Driver c:\windows\system32\drivers\dmio.sys Kernel Driver
Yes Boot Running OK Normal No Yes
dmload dmload c:\windows\system32\drivers\dmload.sys Kernel Driver Yes Boot
Running OK Normal No Yes
dpti2o dpti2o Not Available Kernel Driver No Disabled Stopped OK
Normal No No
elxstor elxstor Not Available Kernel Driver No Disabled Stopped OK
Normal No No
fastfat Fastfat c:\windows\system32\drivers\fastfat.sys File System Driver No
Disabled Stopped OK Normal No No

fdc Floppy Disk Controller Driver c:\windows\system32\drivers\fdc.sys Kernel Driver
Yes Manual Running OK Normal No Yes
fips Fips c:\windows\system32\drivers\fips.sys Kernel Driver Yes System
Running OK Normal No Yes
flpydisk Floppy Disk Driver c:\windows\system32\drivers\flpydisk.sys Kernel Driver
Yes Manual Running OK Normal No Yes
fltmgr FltMgr c:\windows\system32\drivers\fltmgr.sys File System Driver Yes Boot
Running OK Normal No Yes
ftdisk Volume Manager Driver c:\windows\system32\drivers\ftdisk.sys Kernel Driver Yes
Boot Running OK Normal No Yes
gpc Generic Packet Classifier c:\windows\system32\drivers\msgpc.sys Kernel Driver
Yes Manual Running OK Normal No Yes
hidusb Microsoft HID Class Driver c:\windows\system32\drivers\hidusb.sys Kernel Driver
Yes Manual Running OK Ignore No Yes
hpcisss hpcisss Not Available Kernel Driver No Disabled Stopped OK
Normal No No
http HTTP c:\windows\system32\drivers\http.sys Kernel Driver No Manual
Stopped OK Normal No No
i2omgmt i2omgmt Not Available Kernel Driver No System Stopped
OK Normal No No
i8042prt i8042prt c:\windows\system32\drivers\i8042prt.sys Kernel Driver
No System Stopped OK Ignore No No
iirsp iirsp Not Available Kernel Driver No Disabled Stopped OK
Normal No No
imapi CD-Burning Filter Driver c:\windows\system32\drivers\imapi.sys Kernel Driver No
System Stopped OK Normal No No
intelide IntelIde Not Available Kernel Driver No Disabled Stopped OK
Normal No No
intelppm Intel Processor Driver c:\windows\system32\drivers\intelppm.sys Kernel
Driver Yes Manual Running OK Normal No Yes
ip6fw IPv6 Windows Firewall Driver c:\windows\system32\drivers\ip6fw.sys Kernel Driver
No Manual Stopped OK Normal No No
ipfilterdriver IP Traffic Filter Driver c:\windows\system32\drivers\ipfltdrv.sys Kernel Driver
No Manual Stopped OK Normal No No
ipinip IP in IP Tunnel Driver c:\windows\system32\drivers\ipinip.sys Kernel Driver No
Manual Stopped OK Normal No No
ipnat IP Network Address Translator c:\windows\system32\drivers\ipnat.sys Kernel Driver
No Manual Stopped OK Normal No No
ipsec IPSEC driver c:\windows\system32\drivers\ipsec.sys Kernel Driver Yes System
Running OK Normal No Yes
irenum IR Enumerator Service c:\windows\system32\drivers\irenum.sys Kernel Driver No
Manual Stopped OK Normal No No
isapnp PnP ISA/EISA Bus Driver c:\windows\system32\drivers\isapnp.sys Kernel Driver
Yes Boot Running OK Critical No Yes
kbdclass Keyboard Class Driver c:\windows\system32\drivers\kbdclass.sys Kernel
Driver Yes System Running OK Normal No Yes
kbdhid Keyboard HID Driver c:\windows\system32\drivers\kbdhid.sys Kernel Driver Yes
System Running OK Ignore No Yes
ksecdd KSecDD c:\windows\system32\drivers\ksecdd.sys Kernel Driver Yes Boot
Running OK Normal No Yes
ksthunk Kernel Streaming WOW64 Thunk Service
c:\windows\system32\drivers\ksthunk.sys Kernel Driver Yes Manual
Running OK Normal No Yes
l2nd Broadcom NetXtreme II BXND c:\windows\system32\drivers\bxnd52a.sys Kernel
Driver Yes Manual Running OK Normal No Yes

lp6nds35 lp6nds35 Not Available Kernel Driver No Disabled
Stopped OK Normal No No
mnmdd mnmdd c:\windows\system32\drivers\mnmdd.sys Kernel Driver Yes System
Running OK Ignore No Yes
modem Modem c:\windows\system32\drivers\modem.sys Kernel Driver No Manual
Stopped OK Ignore No No
mouclass Mouse Class Driver c:\windows\system32\drivers\mouclass.sys Kernel
Driver Yes System Running OK Normal No Yes
mouhid Mouse HID Driver c:\windows\system32\drivers\mouhid.sys Kernel Driver
Yes Manual Running OK Ignore No Yes
mountmgr Mount Point Manager c:\windows\system32\drivers\mountmgr.sys Kernel
Driver Yes Boot Running OK Normal No Yes
mraid35x mraid35x Not Available Kernel Driver No Disabled
Stopped OK Normal No No
mrxdav WebDav Client Redirector c:\windows\system32\drivers\mrxdav.sys File
System Driver No Manual Stopped OK Normal No No
mrxsmb MRXSMB c:\windows\system32\drivers\mrxsmb.sys File System Driver
Yes System Running OK Normal No Yes
msfs Msfs c:\windows\system32\drivers\msfs.sys File System Driver Yes System
Running OK Normal No Yes
mssmbios Microsoft System Management BIOS Driver
c:\windows\system32\drivers\mssmbios.sys Kernel Driver Yes Manual
Running OK Normal No Yes
mup Mup c:\windows\system32\drivers\mup.sys File System Driver Yes Boot
Running OK Normal No Yes
ndis NDIS System Driver c:\windows\system32\drivers\ndis.sys Kernel Driver Yes
Boot Running OK Normal No Yes
ndistapi Remote Access NDIS TAPI Driver c:\windows\system32\drivers\ndistapi.sys
Kernel Driver Yes Manual Running OK Normal No Yes
ndisuio NDIS Usermode I/O Protocol c:\windows\system32\drivers\ndisuio.sys Kernel Driver
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 c:\windows\system32\drivers\pci.sys Kernel Driver Yes Boot
Running OK Critical No Yes

pciide PCIIde c:\windows\system32\drivers\pciide.sys Kernel Driver Yes Boot
Running OK Normal No Yes
pcmcia Pcmcia c:\windows\system32\drivers\pcmcia.sys Kernel Driver No Disabled
Stopped OK Normal No No
pdcomp PDCOMP Not Available Kernel Driver No Manual Stopped OK
Ignore No No
pdframe PDFRAME Not Available Kernel Driver No Manual Stopped
OK Ignore No No
pdreli PDRELI Not Available Kernel Driver No Manual Stopped OK
Ignore No No
pdrframe PDRFRAME Not Available Kernel Driver No Manual Stopped
OK Ignore No No
percsas percsas c:\windows\system32\drivers\percsas.sys Kernel Driver Yes Boot
Running OK Normal No Yes
pptpminiport WAN Miniport (PPTP) c:\windows\system32\drivers\raspptp.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 ql2300 Not Available Kernel Driver No Disabled Stopped OK
Normal No No
rasacd Remote Access Auto Connection Driver c:\windows\system32\drivers\rasacd.sys Kernel
Driver Yes System Running OK Normal No Yes
rasl2tp WAN Miniport (L2TP) c:\windows\system32\drivers\rasl2tp.sys Kernel Driver Yes
Manual Running OK Normal No Yes
raspppoe Remote Access PPPOE Driver c:\windows\system32\drivers\raspppoe.sys
Kernel Driver Yes Manual Running OK Normal No Yes
raspti Direct Parallel c:\windows\system32\drivers\raspti.sys Kernel Driver Yes Manual
Running OK Normal No Yes
rdbss Rdbss c:\windows\system32\drivers\rdbss.sys File System Driver Yes System
Running OK Normal No Yes
rdpcdd RDPCDD c:\windows\system32\drivers\rdpcdd.sys Kernel Driver Yes System
Running OK Ignore No Yes
rdpdr Terminal Server Device Redirector Driver c:\windows\system32\drivers\rdpdr.sys
Kernel Driver Yes Manual Running OK Normal No Yes
rdpwd RDPWD c:\windows\system32\drivers\rdpwd.sys Kernel Driver Yes Manual
Running OK Ignore No Yes
redbook Digital CD Audio Playback Filter Driver
c:\windows\system32\drivers\redbook.sys Kernel Driver Yes System
Running OK Normal No Yes
secdrv Security Driver c:\windows\system32\drivers\secdrv.sys Kernel Driver Yes Auto
Running OK Normal No Yes
serenum Serenum Filter Driver c:\windows\system32\drivers\serenum.sys Kernel
Driver Yes Manual Running OK Normal No Yes
serial Serial port driver c:\windows\system32\drivers\serial.sys Kernel Driver Yes
System Running OK Ignore No Yes
sfloppy Sfloppy c:\windows\system32\drivers\sfloppy.sys Kernel Driver No System
Stopped OK Ignore No No
simbad Simbad Not Available Kernel Driver No Disabled Stopped OK
Normal No No
srv Srv c:\windows\system32\drivers\srv.sys File System Driver Yes Manual
Running OK Normal No Yes
swenum Software Bus Driver c:\windows\system32\drivers\swenum.sys Kernel
Driver Yes Manual Running OK Normal No Yes
symc8xx symc8xx Not Available Kernel Driver No Disabled
Stopped OK Normal No No

```
symmpi symmpi Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
sym_hi sym_hi Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
sym_u3 sym_u3 Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
tcpip TCP/IP Protocol Driver c:\windows\system32\drivers\tcpip.sys Kernel Driver Yes
        System Running OK Normal No Yes
tdpipe TDPIPE c:\windows\system32\drivers\tdpipe.sys Kernel Driver No Manual
        Stopped OK Ignore No No
tdtcp TDTCP c:\windows\system32\drivers\tdtcp.sys Kernel Driver Yes Manual
        Running OK Ignore No Yes
termdd Terminal Device Driver c:\windows\system32\drivers\termdd.sys Kernel Driver Yes
        System Running OK Normal No Yes
toside TosIde Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
udfs Udfs c:\windows\system32\drivers\udfs.sys File System Driver No
        Disabled Stopped OK Normal No No
ultra ultra Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
update Microcode Update Driver c:\windows\system32\drivers\update.sys Kernel Driver
        Yes Manual Running OK Normal No Yes
usbccgp Microsoft USB Generic Parent Driver
        c:\windows\system32\drivers\usbccgp.sys Kernel Driver Yes Manual
        Running OK Normal No Yes
usbehci Microsoft USB 2.0 Enhanced Host Controller Miniport Driver
        c:\windows\system32\drivers\usbehci.sys Kernel Driver Yes Manual
        Running OK Normal No Yes
usbhub Microsoft USB Standard Hub Driver c:\windows\system32\drivers\usbhub.sys
        Kernel Driver Yes Manual Running OK Normal No Yes
usbuhci Microsoft USB Universal Host Controller Miniport Driver
        c:\windows\system32\drivers\usbuhci.sys Kernel Driver Yes Manual
        Running OK Normal No Yes
vga vga c:\windows\system32\drivers\vgapnp.sys Kernel Driver Yes Manual
        Running OK Ignore No Yes
vgasave VGA Display Controller. c:\windows\system32\drivers\vga.sys Kernel Driver
        No System Stopped OK Ignore No No
viaide ViaIde Not Available Kernel Driver No Disabled Stopped OK
        Normal No No
volsnap Storage volumes c:\windows\system32\drivers\volsnap.sys Kernel Driver
        Yes Boot Running OK Normal No Yes
wanarp Remote Access IP ARP Driver c:\windows\system32\drivers\wanarp.sys Kernel
Driver Yes Manual Running OK Normal No Yes
wdica WDICA Not Available Kernel Driver No Manual Stopped OK Ignore
        No No
wlbs Network Load Balancing c:\windows\system32\drivers\wlbs.sys Kernel Driver
        No Manual Stopped OK Normal No No

[Signed Drivers]

Device Name Signed Device Class Driver Version Driver Date Manufacturer INF
Name Driver Name Device ID
Microsoft System Management BIOS Driver Yes SYSTEM 5.2.3790.1830
        10/1/2002 (Standard system devices) machine.inf Not Available
        ROOT\SYSTEM\0002
```

Microcode Update Device Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available ROOT\SYSTEM\0001
Plug and Play Software Device Enumerator Yes SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices) machine.inf Not Available
ROOT\SYSTEM\0000
Terminal Server Mouse Driver Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available ROOT\RDP_MOU\0000
Terminal Server Keyboard Driver Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available ROOT\RDP_KBD\0000
Terminal Server Device Redirector Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available ROOT\RDPDR\0000
Direct Parallel Yes NET 5.2.3790.1830 10/1/2002 Microsoft netrasa.inf
Not Available ROOT\MS_PTIMINIPORT\0000
WAN Miniport (PPTP) Yes NET 5.2.3790.1830 10/1/2002 Microsoft
netrasa.inf Not Available ROOT\MS_PPTPMINIPORT\0000
WAN Miniport (PPPOE) Yes NET 5.2.3790.1830 10/1/2002 Microsoft
netrasa.inf Not Available ROOT\MS_PPPOEMINIPORT\0000
WAN Miniport (IP) Yes NET 5.2.3790.1830 10/1/2002 Microsoft
netrasa.inf Not Available ROOT\MS_NDISWANIP\0000
WAN Miniport (L2TP) Yes NET 5.2.3790.1830 10/1/2002 Microsoft
netrasa.inf Not Available ROOT\MS_L2TPMINIPORT\0000
Video Codecs Yes MEDIA 5.2.3790.1830 10/1/2002 (Standard system devices)
wave.inf Not Available ROOT\MEDIA\MS_MMVID
Legacy Video Capture Devices Yes MEDIA 5.2.3790.1830 10/1/2002 (Standard
system devices)wave.inf Not Available ROOT\MEDIA\MS_MMVCD
Media Control Devices Yes MEDIA 5.2.3790.1830 10/1/2002 (Standard system
devices) wave.inf Not Available ROOT\MEDIA\MS_MMMCI
Legacy Audio Drivers Yes MEDIA 5.2.3790.1830 10/1/2002 (Standard system
devices) wave.inf Not Available ROOT\MEDIA\MS_MMDRV
Audio Codecs Yes MEDIA 5.2.3790.1830 10/1/2002 (Standard system devices)
wave.inf Not Available ROOT\MEDIA\MS_MMACM
Remote Access IP ARP Driver Not Available LEGACYDRIVER Not Available Not
Available Not Available Not Available Not Available
ROOT\LEGACY_WANARP\0000
volsnap Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_VOLSNAP\0000
TDTCP Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_TDTCP\0000
TCP/IP Protocol Driver Not Available LEGACYDRIVER Not Available Not Available
Not Available Not Available Not Available ROOT\LEGACY_TCPIP\0000
Security Driver Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_SECDRV\0000
sacdrv Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_SACDRV\0000
RDPWD Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_RDPWD\0000
RDPCDD Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_RDPCDD\0000
Remote Access Auto Connection Driver Not Available LEGACYDRIVER Not Available
Not Available Not Available Not Available Not Available
ROOT\LEGACY_RASACD\0000
Partition Manager Not Available LEGACYDRIVER Not Available Not Available
Not Available Not Available Not Available ROOT\LEGACY_PARTMGR\0000
Null Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_NULL\0000

NetBios over Tcpip Not Available LEGACYDRIVER Not Available Not Available
Not Available Not Available Not Available ROOT\LEGACY_NETBT\0000
NDProxy Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_NDPROXY\0000
NDIS Usermode I/O Protocol Not Available LEGACYDRIVER Not Available Not
Available Not Available Not Available Not Available
ROOT\LEGACY_NDISUIO\0000
Remote Access NDIS TAPI Driver Not Available LEGACYDRIVER Not Available
Not Available Not Available Not Available Not Available
ROOT\LEGACY_NDISTAPI\0000
NDIS System Driver Not Available LEGACYDRIVER Not Available Not Available
Not Available Not Available Not Available ROOT\LEGACY_NDIS\0000
mountmgr Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_MOUNTMGR\0000
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
IP Network Address Translator Not Available LEGACYDRIVER Not Available Not
Available Not Available Not Available Not Available ROOT\LEGACY_IPNAT\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
CdaD10BA Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_CDAD10BA\0000
CdaC15BA Not Available LEGACYDRIVER Not Available Not Available Not
Available Not Available Not Available ROOT\LEGACY_CDAC15BA\0000
Beep Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_BEEP\0000
AFD Not Available LEGACYDRIVER Not Available Not Available Not Available
Not Available Not Available ROOT\LEGACY_AFD\0000
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE427D5E7OFFSET2CBE8C8800LEN
GTHCD91089800
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE427D5E7OFFSET1DFF826000LEN
GTHEBF0A2800
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE427D5E7OFFSET7E00LENGTH1DF
F81E200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available

STORAGE\VOLUME\1&30A96598&0&SIGNATUREAADFAADFOFFSET27FCBB200LENGTH859FBE1C00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATUREAADFAADFOFFSET7E00LENGTH27FCB3400
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATURE41EF9D8OFFSET24D9AD5400LENGTHD575E7CC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATURE41EF9D8OFFSET18B46AC800LENGTHC25428C00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATURE41EF9D8OFFSET7E00LENGTH18B46A4A00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATURE41EF9DAOFFSET18B46AC800LENGTHC25428C00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft volume.inf Not Available STORAGE\VOLUME\1&30A96598&0&SIGNATURE41EF9DAOFFSET7E00LENGTH18B46A4A00
Volume Manager Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)machine.inf Not Available ROOT\FTDISK\0000
Logical Disk Manager Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)machine.inf Not Available ROOT\DMIO\0000
ACPI Fixed Feature Button Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\FIXEDBUTTON\2&DABA3FF&0
High precision event timer Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0103\0
Motherboard resources Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)machine.inf Not Available ACPI\PNP0C02\0
Secondary IDE Channel Yes HDC 5.2.3790.1830 10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available PCIIDE\IDECHANNEL\4&1D8A9C03&0&1
CD-ROM Drive Yes CDROM 5.2.3790.1830 10/1/2002 (Standard CD-ROM drives) cdrom.inf Not Available IDE\CDROMLITE-ON_CD-ROM_LTN-4891S_____NDS3____\5&41A3CB2&0&0.0.0
Primary IDE Channel Yes HDC 5.2.3790.1830 10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available PCIIDE\IDECHANNEL\4&1D8A9C03&0&0
Standard Dual Channel PCI IDE Controller Yes HDC 5.2.3790.1830 10/1/2002 (Standard IDE ATA/ATAPI controllers) mshdc.inf Not Available PCI\VEN_8086&DEV_269E&SUBSYS_01B11028&REV_09\3&61AAA01&0&F9
System board Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\IPI0001\5
System board Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0C01\0
Communications Port Yes PORTS 5.2.3790.1830 10/1/2002 (Standard port types) msports.inf Not Available ACPI\PNP0501\1

Floppy disk drive Yes FLOPPYDISK 5.2.3790.1830 10/1/2002 (Standard floppy disk drives) flpydisk.inf Not Available FDC\GENERIC_FLOPPY_DRIVE\5&33C0F973&0&0
Standard floppy disk controller Yes FDC 5.2.3790.1830 10/1/2002 (Standard floppy disk controllers) fdc.inf Not Available ACPI\PNP0700\4&2AA4AD3D&0
System timer Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0100\4&2AA4AD3D&0
System CMOS/real time clock Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0B00\4&2AA4AD3D&0
System board Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0C01\2
Programmable interrupt controller Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0000\4&2AA4AD3D&0
Numeric data processor Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0C04\4&2AA4AD3D&0
Direct memory access controller Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available ACPI\PNP0200\4&2AA4AD3D&0
PCI standard ISA bridge Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices) machine.inf Not Available PCI\VEN_8086&DEV_2670&SUBSYS_00000000&REV_09\3&61AAA01&0&F8
Plug and Play Monitor Yes MONITOR 5.2.3790.1830 10/1/2002 (Standard monitor types) monitor.inf Not Available DISPLAY\DEL3580\5&EEED524&0&12345678&16&0D
Standard VGA Graphics Adapter Yes DISPLAY 5.2.3790.1830 10/1/2002 (Standard display types) display.inf Not Available PCI\VEN_1002&DEV_515E&SUBSYS_01B11028&REV_02\4&2014205D&0&68F0
Intel(R) 82801 PCI Bridge - 244E Yes SYSTEM 5.2.3790.1830 10/1/2002 Intel machine.inf Not Available PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_D9\3&61AAA01&0&F0
Generic USB Hub Yes USB 5.2.3790.1830 10/1/2002 (Generic USB Hub) usb.inf Not Available USB\VID_04B4&PID_6560\5&6F526B7&0&7
USB Root Hub Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available USB\ROOT_HUB20\4&25F3EE70&0
Standard Enhanced PCI to USB Host Controller Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available PCI\VEN_8086&DEV_268C&SUBSYS_01B11028&REV_09\3&61AAA01&0&EF
USB Root Hub Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available USB\ROOT_HUB\4&1C386FEF&0
Standard Universal PCI to USB Host Controller Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available PCI\VEN_8086&DEV_268B&SUBSYS_01B11028&REV_09\3&61AAA01&0&EB
USB Root Hub Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available USB\ROOT_HUB\4&2DD355BC&0
Standard Universal PCI to USB Host Controller Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usbport.inf Not Available PCI\VEN_8086&DEV_268A&SUBSYS_01B11028&REV_09\3&61AAA01&0&EA
HID-compliant mouse Yes MOUSE 5.2.3790.1830 10/1/2002 Microsoft msmouse.inf Not Available HID\VID_0557&PID_2221&MI_01\7&7EC711E&0&0000
USB Human Interface Device Yes HIDCLASS 5.2.3790.1830 10/1/2002 (Standard system devices) input.inf Not Available USB\VID_0557&PID_2221&MI_01\6&1D334AC&0&0001

HID Keyboard Device Yes KEYBOARD 5.2.3790.1830 10/1/2002 (Standard keyboards) keyboard.inf Not Available
HID\VID_0557&PID_2221&MI_00\7&1B1C8D5C&0&0000
USB Human Interface Device Yes HIDCLASS 5.2.3790.1830 10/1/2002
(Standard system devices) input.inf Not Available
USB\VID_0557&PID_2221&MI_00\6&1D334AC&0&0000
USB Composite Device Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller) usb.inf Not Available USB\VID_0557&PID_2221\5&3014DA89&0&1
USB Root Hub Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller)
usbport.inf Not Available USB\ROOT_HUB\4&2DC298A6&0
Standard Universal PCI to USB Host Controller Yes USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf Not Available
PCI\VEN_8086&DEV_2689&SUBSYS_01B11028&REV_09\3&61AAA01&0&E9
USB Root Hub Yes USB 5.2.3790.1830 10/1/2002 (Standard USB Host Controller)
usbport.inf Not Available USB\ROOT_HUB\4&1C492D05&0
Standard Universal PCI to USB Host Controller Yes USB 5.2.3790.1830 10/1/2002
(Standard USB Host Controller) usbport.inf Not Available
PCI\VEN_8086&DEV_2688&SUBSYS_01B11028&REV_09\3&61AAA01&0&E8
Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client) Yes NET 2.6.14.0
4/3/2006 Broadcom Corporation oem3.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_01B11028&REV_12\6&2F68317E&0&2005 0500
Broadcom BCM5708C NetXtreme II GigE Yes SYSTEM 2.6.17.0
4/21/2006 Broadcom Corporation oem5.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_01B11028&REV_12\5&43097C6&0&0000E0
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C3\4&187919FE&0&00E0
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_2690&SUBSYS_00000000&REV_09\3&61AAA01&0&E0
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F6&SUBSYS_00000000&REV_12\3&61AAA01&0&B0
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F5&SUBSYS_00000000&REV_12\3&61AAA01&0&A8
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F3&SUBSYS_00000000&REV_12\3&61AAA01&0&98
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F1&SUBSYS_00000000&REV_12\3&61AAA01&0&88
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_12\3&61AAA01&0&82
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_12\3&61AAA01&0&81
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_12\3&61AAA01&0&80
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25E7&SUBSYS_00000000&REV_12\3&61AAA01&0&38

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0372&SUBSYS_00000000&REV_00\4&149DD53B&0&0230
DELL PERC RAID Virtual Device No SYSTEM 1.20.0.64 12/9/2005
DELL oem0.inf Not Available
SCSI\OTHER&VEN___RAID&PROD__DUMMYDEVICE&REV_0001\6&1BE1250F&0&1
400
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_DELL&PROD_PERC_5/E_ADAPTER&REV_1.00\6&1BE1250F&0&10
0
DELL PERC 5/E Adapter RAID Controller No SCSIADAPTER 1.20.0.64
12/9/2005 DELL oem1.inf Not Available
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&376DDE58&0&700030
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0370&SUBSYS_00000000&REV_00\4&149DD53B&0&0030
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25F9&SUBSYS_00000000&REV_12\3&61AAA01&0&30
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0372&SUBSYS_00000000&REV_00\4&28CEA4F8&0&0228
DELL PERC RAID Virtual Device No SYSTEM 1.20.0.64 12/9/2005
DELL oem0.inf Not Available
SCSI\OTHER&VEN___RAID&PROD__DUMMYDEVICE&REV_0001\6&107D05B&0&14
00
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_DELL&PROD_PERC_5/I&REV_1.00\6&107D05B&0&100
DELL PERC 5/i Integrated RAID Controller No SCSIADAPTER 1.20.0.64
12/9/2005 DELL oem1.inf Not Available
PCI\VEN_1028&DEV_0015&SUBSYS_1F031028&REV_00\5&22FD9970&0&700028
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0370&SUBSYS_00000000&REV_00\4&28CEA4F8&0&0028
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25E5&SUBSYS_00000000&REV_12\3&61AAA01&0&28
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0372&SUBSYS_00000000&REV_00\4&201440C3&0&0220
DELL PERC RAID Virtual Device No SYSTEM 1.20.0.64 12/9/2005
DELL oem0.inf Not Available
SCSI\OTHER&VEN___RAID&PROD__DUMMYDEVICE&REV_0001\6&2DAC0A12&0&1
400
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_DELL&PROD_PERC_5/E_ADAPTER&REV_1.00\6&2DAC0A12&0&1
00
DELL PERC 5/E Adapter RAID Controller No SCSIADAPTER 1.20.0.64
12/9/2005 DELL oem1.inf Not Available
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&20524F73&0&700020

PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0370&SUBSYS_00000000&REV_00\4&201440C3&0&0020
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25E4&SUBSYS_00000000&REV_12\3&61AAA01&0&20
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0372&SUBSYS_00000000&REV_00\4&3646D6F4&0&0218
DELL PERC RAID Virtual Device No SYSTEM 1.20.0.64 12/9/2005
DELL oem0.inf Not Available
SCSI\OTHER&VEN___RAID&PROD__DUMMYDEVICE&REV_0001\6&1037D935&0&1
400
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available
SCSI\DISK&VEN_DELL&PROD_PERC_5/E_ADAPTER&REV_1.03\6&1037D935&0&10
0
DELL PERC 5/E Adapter RAID Controller No SCSIADAPTER 1.20.0.64
12/9/2005 DELL oem1.inf Not Available
PCI\VEN_1028&DEV_0015&SUBSYS_1F011028&REV_00\5&1E758DE3&0&700018
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_0370&SUBSYS_00000000&REV_00\4&3646D6F4&0&0018
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25E3&SUBSYS_00000000&REV_12\3&61AAA01&0&18
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_350C&SUBSYS_00000000&REV_01\4&3667122&0&0310
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3514&SUBSYS_00000000&REV_01\5&2D9FD309&0&080010
Broadcom BCM5708C NetXtreme II GigE (NDIS VBD Client) Yes NET 2.6.14.0
4/3/2006 Broadcom Corporation oem3.inf Not Available
B06BDRV\L2ND&PCI_164C14E4&SUBSYS_01B11028&REV_12\8&126A2D63&0&2005
0900
Broadcom BCM5708C NetXtreme II GigE Yes SYSTEM 2.6.17.0
4/21/2006 Broadcom Corporation oem5.inf Not Available
PCI\VEN_14E4&DEV_164C&SUBSYS_01B11028&REV_12\7&2A8A8A3B&0&00000000
10
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_1166&DEV_0103&SUBSYS_00000000&REV_C3\6&6336DE7&0&00000010
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3510&SUBSYS_00000000&REV_01\5&2D9FD309&0&000010
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_3500&SUBSYS_00000000&REV_01\4&3667122&0&0010
PCI standard PCI-to-PCI bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25E2&SUBSYS_00000000&REV_12\3&61AAA01&0&10
PCI standard host CPU bridge Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available
PCI\VEN_8086&DEV_25C0&SUBSYS_00000000&REV_12\3&61AAA01&0&00

PCI busYes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)
machine.inf Not Available ACPI\PNP0A03\2&DABA3FF&0
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-_EM64T_FAMILY_6_MODEL_15_3
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-_EM64T_FAMILY_6_MODEL_15_2
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-_EM64T_FAMILY_6_MODEL_15_1
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-_EM64T_FAMILY_6_MODEL_15_0
Microsoft ACPI-Compliant System Yes SYSTEM 5.2.3790.1830 10/1/2002
Microsoft acpi.inf Not Available ACPI_HAL\PNP0C08\0
ACPI Multiprocessor x64-based PC Yes COMPUTER 5.2.3790.1830 10/1/2002
(Standard computers) hal.inf Not Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not Available Not Available Not Available
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_CHECKNO <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
OS Windows_NT <SYSTEM>
Path
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\Progra
m Files (x86)\Microsoft SQL Server\80\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\90\Tools\binn\;C:\Program Files (x86)\Microsoft SQL Server\90\Tools\binn\;C:\Program
Files\Microsoft SQL Server\90\DTS\Binn\;.;C:\Program Files\Microsoft SQL
Server\MSSQL.1\MSSQL\Binn <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
PROCESSOR_ARCHITECTURE AMD64 <SYSTEM>
PROCESSOR_IDENTIFIER EM64T Family 6 Model 15 Stepping 7, GenuineIntel
<SYSTEM>
PROCESSOR_LEVEL 6 <SYSTEM>
PROCESSOR_REVISION 0f07 <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 PE2900\Administrator
TMP %USERPROFILE%\Local Settings\Temp PE2900\Administrator

[Print Jobs]

Document Size Owner Notify Status Time Submitted Start Time Until Time
Elapsed Time Pages Printed Job ID Priority Parameters Driver Print Processor
Host Print Queue Data Type Name

[Network Connections]

Local Name Remote Name Type Status User Name

[Running Tasks]

Name Path Process ID Priority Min Working Set Max Working Set Start Time Version Size File Date
system idle process Not Available 0 0 Not Available Not Available Not Available Not Available Not Available Not Available
system Not Available 4 8 0 1413120 Not Available Not Available Not Available Not Available
smss.exe Not Available 364 11 204800 1413120 2/25/2007 11:34 AM Not Available Not Available Not Available
csrss.exe Not Available 600 13 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe 672 13 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 901.00 KB (922,624 bytes) 3/25/2005 6:00 AM
services.exe c:\windows\system32\services.exe 716 9 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 216.50 KB (221,696 bytes) 3/25/2005 6:00 AM
lsass.exe c:\windows\system32\lsass.exe 728 9 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 3/25/2005 6:00 AM
svchost.exe c:\windows\system32\svchost.exe 932 8 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 6:00 AM
svchost.exe Not Available 1004 8 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
svchost.exe Not Available 212 8 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
svchost.exe Not Available 388 8 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe 424 8 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 6:00 AM
msdtc.exe Not Available 1116 8 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe 1264 8 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 6:00 AM
svchost.exe Not Available 1320 8 Not Available Not Available 2/25/2007 11:34 AM Not Available Not Available Not Available
svchost.exe c:\windows\system32\svchost.exe 1716 8 204800 1413120 2/25/2007 11:34 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 3/25/2005 6:00 AM
wmiprvse.exe Not Available 308 8 Not Available Not Available 2/25/2007 11:35 AM Not Available Not Available Not Available
csrss.exe Not Available 1848 13 Not Available Not Available 2/25/2007 12:04 PM Not Available Not Available Not Available
winlogon.exe c:\windows\system32\winlogon.exe 1864 13 204800 1413120 2/25/2007 12:04 PM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 901.00 KB (922,624 bytes) 3/25/2005 6:00 AM

rdpclip.exe c:\windows\system32\rdpclip.exe 620 8 204800 1413120 2/25/2007 12:04 PM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 99.00 KB (101,376 bytes) 2/23/2007 12:45 PM
explorer.exe c:\windows\explorer.exe 1036 8 204800 1413120 2/25/2007 12:04 PM 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.30 MB (1,364,480 bytes) 3/25/2005 6:00 AM
logon.scr Not Available 1428 4 Not Available Not Available 2/25/2007 12:14 PM Not Available Not Available Not Available
cmd.exe c:\windows\system32\cmd.exe 384 8 204800 1413120 2/26/2007 11:13 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 538.50 KB (551,424 bytes) 3/25/2005 6:00 AM
sqlservr.exe c:\program files\microsoft sql server\mssql.1\mssql\binn\sqlservr.exe 1156 13 204800 1413120 2/26/2007 11:13 AM 2005.090.1399.00 37.56 MB (39,379,672 bytes) 10/14/2005 3:37 PM
helpctr.exe c:\windows\pchealth\helpctr\binaries\helpctr.exe 1432 8 204800 1413120 2/26/2007 11:21 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.30 MB (1,363,456 bytes) 2/23/2007 12:47 PM
wmiprvse.exe Not Available 644 8 Not Available Not Available 2/26/2007 11:21 AM Not Available Not Available Not Available
helpsvc.exe c:\windows\pchealth\helpctr\binaries\helpsvc.exe 1628 8 204800 1413120 2/26/2007 11:21 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.52 MB (1,591,296 bytes) 2/23/2007 12:47 PM

[Loaded Modules]

Name Version Size File Date Manufacturer Path
winlogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 901.00 KB (922,624 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winlogon.exe
ntdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.20 MB (1,257,472 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntdll.dll
kernel32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.43 MB (1,500,160 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\kernel32.dll
advapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.00 MB (1,051,136 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\advapi32.dll
rpcrt4 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.63 MB (1,714,176 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rpcrt4.dll
crypt32 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 1.36 MB (1,428,992 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\crypt32.dll
msasn1 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 152.50 KB (156,160 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msasn1.dll
msvcrt 7.0.3790.1830 (srv03_sp1_rtm.050324-1447) 508.00 KB (520,192 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msvcrt.dll
user32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.04 MB (1,085,952 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\user32.dll
gdi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 592.00 KB (606,208 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\gdi32.dll
nddeapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 25.00 KB (25,600 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\nddeapi.dll
profmap 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.00 KB (36,864 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\profmap.dll
netapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 589.00 KB (603,136 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netapi32.dll
userenv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.02 MB (1,069,056 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\userenv.dll

psapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 29.00 KB (29,696 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\psapi.dll
regapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 108.50 KB (111,104 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\regapi.dll
secur32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 120.00 KB (122,880 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\secur32.dll
setupapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.45 MB (1,523,200 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\setupapi.dll
version 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 28.00 KB (28,672 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\version.dll
winsta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 89.00 KB (91,136 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winsta.dll
ws2_32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 176.50 KB (180,736 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ws2_32.dll
ws2help 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 30.50 KB (31,232 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ws2help.dll
msgina 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.14 MB (1,193,472 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msgina.dll
shsvcs 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 193.50 KB (198,144 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\shsvcs.dll
shlwapi 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 606.50 KB (621,056 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\shlwapi.dll
sfc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 6.00 KB (6,144 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\sfc.dll
sfc_os 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 183.50 KB (187,904 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\sfc_os.dll
wintrust 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 297.50 KB (304,640 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wintrust.dll
imagehlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 57.50 KB (58,880 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\imagehlp.dll
ole32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.43 MB (2,543,616 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ole32.dll
comctl32 6.0 (srv03_sp1_rtm.050324-1447) 1.51 MB (1,584,128 bytes)
2/23/2007 6:37 AM Microsoft Corporation
c:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.3790.1830_x-ww_aced72af\comctl32.dll
winscard 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 230.00 KB (235,520 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winscard.dll
wtsapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 29.00 KB (29,696 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wtsapi32.dll
sxs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.91 MB (2,003,968 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\sxs.dll
shell32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 10.01 MB (10,492,416 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\shell32.dll
rsaenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 241.96 KB (247,768 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rsaenh.dll
wldap32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 390.00 KB (399,360 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wldap32.dll
cscdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 151.50 KB (155,136 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cscdll.dll
dimsntfy 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 28.00 KB (28,672 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\dimsntfy.dll
wlnotify 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 148.00 KB (151,552 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wlnotify.dll

mpr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 115.00 KB (117,760 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mpr.dll
oleaut32 5.2.3790.1830 1.06 MB (1,116,160 bytes) 3/25/2005 6:00 AM
Microsoft Corporation c:\windows\system32\oleaut32.dll
winmm 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 303.50 KB (310,784 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winmm.dll
winspool 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 247.00 KB (252,928 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winspool.drv
comctl32 5.82 (srv03_sp1_rtm.050324-1447) 934.50 KB (956,928 bytes)
2/23/2007 6:37 AM Microsoft Corporation
c:\windows\winsxs\amd64_microsoft.windows.common-
controls_6595b64144ccf1df_5.82.3790.1830_x-ww_4d792d2a\comctl32.dll
uxtheme 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 494.50 KB (506,368 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\uxtheme.dll
scredir 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 38.50 KB (39,424 bytes)3/25/2005 6:00
AM Microsoft Corporation c:\windows\system32\scredir.dll
samlib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 69.00 KB (70,656 bytes)3/25/2005 6:00
AM Microsoft Corporation c:\windows\system32\samlib.dll
clbcatq 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 865.00 KB (885,760 bytes)
2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\clbcatq.dll
comres 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 779.50 KB (798,208 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\comres.dll
cscui 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 441.00 KB (451,584 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cscui.dll
rdpsnd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 25.00 KB (25,600 bytes)3/25/2005 6:00
AM Microsoft Corporation c:\windows\system32\rdpsnd.dll
msacm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 31.00 KB (31,744 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msacm32.drv
msacm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 112.00 KB (114,688 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msacm32.dll
imaadp32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\imaadp32.acm
msadp32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 23.50 KB (24,064 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msadp32.acm
msg711 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 13.50 KB (13,824 bytes)3/25/2005 6:00
AM Microsoft Corporation c:\windows\system32\msg711.acm
msgsm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 34.50 KB (35,328 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msgsm32.acm
tssoft32 1.01 13.50 KB (13,824 bytes)3/25/2005 6:00 AM DSP GROUP, INC.
c:\windows\system32\tssoft32.acm
tsd32 1.03 24.50 KB (25,088 bytes)3/25/2005 6:00 AM DSP GROUP, INC.
c:\windows\system32\tsd32.dll
ntmarta 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 222.50 KB (227,840 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntmarta.dll
xpsp2res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.77 MB (2,899,456 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\xpsp2res.dll
drprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes)3/25/2005 6:00
AM Microsoft Corporation c:\windows\system32\drprov.dll
ntlanman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 71.50 KB (73,216 bytes)
3/25/2005 6:00 AM 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 6:00 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 6:00 AM Microsoft Corporation c:\windows\system32\netui1.dll

davclnt 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 38.00 KB (38,912 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\davclnt.dll
wbemprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 38.00 KB (38,912 bytes)
2/23/2007 12:44 PM Microsoft Corporation
c:\windows\system32\wbem\wbemprox.dll
wbemcomn 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 524.00 KB (536,576 bytes)
3/25/2005 6:00 AM Microsoft Corporation
c:\windows\system32\wbem\wbemcomn.dll
wbemsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 58.00 KB (59,392 bytes)
2/23/2007 12:44 PM Microsoft Corporation
c:\windows\system32\wbem\wbemsvc.dll
fastprox 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 866.50 KB (887,296 bytes)
2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\fastprox.dll
msvcp60 7.0.3790.1830 (srv03_sp1_rtm.050324-1447) 919.50 KB (941,568 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msvcp60.dll
ntdsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 127.50 KB (130,560 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntdsapi.dll
dnsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 297.50 KB (304,640 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\dnsapi.dll
services 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 216.50 KB (221,696 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\services.exe
ncobjapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 80.00 KB (81,920 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ncobjapi.dll
scesrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 594.50 KB (608,768 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\scesrv.dll
authz 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 167.00 KB (171,008 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\authz.dll
umpnpmgr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 205.00 KB (209,920 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\umpnpmgr.dll
eventlog 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 127.00 KB (130,048 bytes)
3/25/2005 6:00 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\lsass.exe
lsasrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.50 MB (1,568,256 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\lsasrv.dll
samsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.01 MB (1,059,328 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\samsrv.dll
cryptdll 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cryptdll.dll
msprivs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.50 KB (48,640 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msprivs.dll
kerberos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 698.00 KB (714,752 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\kerberos.dll
msv1_0 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 253.00 KB (259,072 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msv1_0.dll
iphlpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 177.00 KB (181,248 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\iphlpapi.dll
netlogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 666.00 KB (681,984 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netlogon.dll
w32time 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 400.50 KB (410,112 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\w32time.dll
schannel 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 248.00 KB (253,952 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\schannel.dll
wdigest 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 130.50 KB (133,632 bytes)
3/25/2005 6:00 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\rassfm.dll
kdcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 409.00 KB (418,816 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\kdcsvc.dll
ntdsa 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.81 MB (2,948,096 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntdsa.dll
esent 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.26 MB (2,366,976 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\esent.dll
ntdsatq 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 51.00 KB (52,224 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntdsatq.dll
mswsock 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 478.00 KB (489,472 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mswsock.dll
scecli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 308.00 KB (315,392 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\scecli.dll
ws03res 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 794.00 KB (813,056 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ws03res.dll
ipsecsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 358.50 KB (367,104 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ipsecsvc.dll
oakley 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 372.50 KB (381,440 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\oakley.dll
winipsec 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 52.50 KB (53,760 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winipsec.dll
pstorsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.00 KB (36,864 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\pstorsvc.dll
psbase 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 124.00 KB (126,976 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\psbase.dll
hnetcfg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 561.00 KB (574,464 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\hnetcfg.dll
wshtcpip 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 29.00 KB (29,696 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wshtcpip.dll
dssenh 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 226.96 KB (232,408 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\dssenh.dll
wlbsctrl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 137.50 KB (140,800 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wlbsctrl.dll
svchost 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\svchost.exe
rpcss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 672.00 KB (688,128 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rpcss.dll
schedsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 308.50 KB (315,904 bytes)
2/23/2007 12:47 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 6:00 AM Microsoft Corporation c:\windows\system32\msidle.dll
wkssvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 221.00 KB (226,304 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wkssvc.dll
wiarpc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 57.00 KB (58,368 bytes)3/25/2005 6:00 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\aelupsvc.dll
apphelp5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 241.00 KB (246,784 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\apphelp.dll
cryptsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 114.00 KB (116,736 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cryptsvc.dll
certcli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 372.00 KB (380,928 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\certcli.dll

atl 3.05.2284 96.50 KB (98,816 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\atl.dll
vssapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.26 MB (1,320,960 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\vssapi.dll
dmserver 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.50 KB (37,376 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\dmserver.dll
es 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 357.00 KB (365,568 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\es.dll
srvsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 156.50 KB (160,256 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\srvsvc.dll
sacsvr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 16.50 KB (16,896 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\sacsvr.dll
seclogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 27.50 KB (28,160 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\seclogon.dll
sens 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 63.50 KB (65,024 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\sens.dll
trkwks 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 177.50 KB (181,760 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\trkwks.dll
wmisvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 227.00 KB (232,448 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\wmisvc.dll
comsvcs 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 2.06 MB (2,156,544 bytes) 2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\comsvcs.dll
browser5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 125.50 KB (128,512 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\browser.dll
netrap 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 26.00 KB (26,624 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netrap.dll
wbemcore 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.24 MB (1,299,968 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 626.50 KB (641,536 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\esscli.dll
wmiutils5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 171.00 KB (175,104 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 353.50 KB (361,984 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 743.00 KB (760,832 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 532.50 KB (545,280 bytes) 2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\wbemess.dll
ncprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 73.00 KB (74,752 bytes)2/23/2007 12:44 PM Microsoft Corporation c:\windows\system32\wbem\ncprov.dll
netman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 457.00 KB (467,968 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netman.dll
mprapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 154.50 KB (158,208 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mprapi.dll
activeds 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 348.50 KB (356,864 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\activeds.dll
adsldpc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 240.50 KB (246,272 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\adsldpc.dll
credui 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 202.00 KB (206,848 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\credui.dll
rtutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 66.00 KB (67,584 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rtutils.dll

netshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 2.32 MB (2,437,120 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netshell.dll
clusapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 127.00 KB (130,048 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\clusapi.dll
rasapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 410.00 KB (419,840 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rasapi32.dll
rasman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 95.50 KB (97,792 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rasman.dll
tapi32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 332.50 KB (340,480 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\tapi32.dll
wininet 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.13 MB (1,186,304 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wininet.dll
wzcsapi5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 49.00 KB (50,176 bytes)3/24/2005 11:35 AM Microsoft Corporation c:\windows\system32\wzcsapi.dll
wzcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 492.00 KB (503,808 bytes)
3/24/2005 11:35 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\wmi.dll
dhcpcsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 219.00 KB (224,256 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\dhcpcsvc.dll
rasdlg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 859.50 KB (880,128 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rasdlg.dll
rasadhlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 12.00 KB (12,288 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\rasadhlp.dll
netcfgx 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.29 MB (1,354,240 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\netcfgx.dll
pchsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 76.00 KB (77,824 bytes)2/23/2007 12:47 PM Microsoft Corporation c:\windows\pchealth\helpctr\binaries\pchsvc.dll
wbemcons 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 65.50 KB (67,072 bytes)
2/23/2007 12:44 PM Microsoft Corporation
c:\windows\system32\wbem\wbemcons.dll
ersvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 31.00 KB (31,744 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ersvc.dll
termsrv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 354.50 KB (363,008 bytes)
2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\termsrv.dll
icaapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 27.50 KB (28,160 bytes)2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\icaapi.dll
mstlsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 187.00 KB (191,488 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mstlsapi.dll
rdpwsx 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 170.13 KB (174,216 bytes)
2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\rdpwsx.dll
rdpclip 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 99.00 KB (101,376 bytes)
2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\rdpclip.exe
wsock32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\wsock32.dll
urlmon 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.02 MB (1,074,176 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\urlmon.dll
explorer 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.30 MB (1,364,480 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\explorer.exe
browseui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 1.53 MB (1,601,536 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\browseui.dll
shdocvw 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 2.30 MB (2,416,128 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\shdocvw.dll
cryptui 5.131.3790.1830 (srv03_sp1_rtm.050324-1447) 705.50 KB (722,432 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cryptui.dll

themeui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 530.50 KB (543,232 bytes)
3/25/2005 6:00 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\msimg32.dll
linkinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 30.00 KB (30,720 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\linkinfo.dll
ntshrui 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 184.00 KB (188,416 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\ntshrui.dll
msi 3.1.4000.1830 4.27 MB (4,476,416 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msi.dll
webcheck 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 439.00 KB (449,536 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\webcheck.dll
stobject 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 142.50 KB (145,920 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\stobject.dll
batmeter 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 41.50 KB (42,496 bytes)
3/25/2005 6:00 AM 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 6:00 AM Microsoft Corporation c:\windows\system32\powrprof.dll
browselc 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 63.00 KB (64,512 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\browselc.dll
mlang 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 686.00 KB (702,464 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mlang.dll
shdoclc 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 589.50 KB (603,648 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\shdoclc.dll
cmd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 538.50 KB (551,424 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\cmd.exe
sqlservr 2005.090.1399.00 37.56 MB (39,379,672 bytes) 10/14/2005 3:37 PM
Microsoft Corporation c:\program files\microsoft sql server\mssql.1\mssql\binn\sqlservr.exe
msvcr80 8.00.50727.42 803.50 KB (822,784 bytes) 9/23/2005 12:26 AM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.42_x-ww_3fea50ad\msvcr80.dll
msvcp80 8.00.50727.42 1.05 MB (1,097,728 bytes) 9/23/2005 12:28 AM
Microsoft Corporation
c:\windows\winsxs\amd64_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.42_x-ww_3fea50ad\msvcp80.dll
opends60 2005.090.1399.00 22.21 KB (22,744 bytes)10/14/2005 3:31 PM
Microsoft Corporation c:\program files\microsoft sql server\mssql.1\mssql\binn\opends60.dll
instapi 2005.090.1399.00 40.71 KB (41,688 bytes)10/14/2005 3:23 PM Microsoft Corporation c:\program files\microsoft sql server\90\shared\instapi.dll
sqlevn70 2005.090.1399.00 1.57 MB (1,642,712 bytes) 10/14/2005 3:32 PM
Microsoft Corporation c:\program files\microsoft sql server\mssql.1\mssql\binn\resources\1033\sqlevn70.rll
sqlos 2005.090.1399.00 15.71 KB (16,088 bytes)10/14/2005 3:35 PM Microsoft Corporation c:\program files\microsoft sql server\mssql.1\mssql\binn\sqlos.dll
mscoree 2.0.50727.42 (RTM.050727-4200) 441.00 KB (451,584 bytes)
9/23/2005 12:37 AM Microsoft Corporation c:\windows\system32\mscoree.dll
xolehlp 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 10.50 KB (10,752 bytes)
2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\xolehlp.dll
msdtcprx 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 805.50 KB (824,832 bytes) 2/23/2007 12:45 PM Microsoft Corporation c:\windows\system32\msdtcprx.dll
mtxclu 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 141.50 KB (144,896 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mtxclu.dll

resutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 98.50 KB (100,864 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\resutils.dll
winrnr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 30.00 KB (30,720 bytes)3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\winrnr.dll
security 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 6.00 KB (6,144 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\security.dll
msfte 12.0.5626.1 3.63 MB (3,803,344 bytes) 8/26/2005 6:17 PM Microsoft Corporation c:\program files\microsoft sql server\mssql.1\mssql\binn\msfte.dll
dbghelp 6.4.0004.3 (vbl_core(jshay).041001-1326) 1.37 MB (1,434,328 bytes)
10/14/2005 3:25 PM Microsoft Corporation c:\program files\microsoft sql server\90\shared\dbghelp.dll
sqlncli 2005.090.1399.00 3.01 MB (3,152,144 bytes) 10/14/2005 3:37 PM
Microsoft Corporation c:\windows\system32\sqlncli.dll
comdlg32 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 446.50 KB (457,216 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\comdlg32.dll
sqlnclir 2005.090.1399.00 201.21 KB (206,040 bytes) 10/14/2005 3:31 PM
Microsoft Corporation c:\windows\system32\sqlnclir.rll
helpctr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.30 MB (1,363,456 bytes)
2/23/2007 12:47 PM 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)
2/23/2007 12:47 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\hcappres.dll
itss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 208.00 KB (212,992 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\itss.dll
msxml3 8.70.1104.0 2.04 MB (2,141,184 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msxml3.dll
pchshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 155.00 KB (158,720 bytes)
2/23/2007 12:47 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\pchshell.dll
mshtml 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 5.65 MB (5,928,448 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mshtml.dll
msls31 3.10.349.0 357.00 KB (365,568 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msls31.dll
msimtf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 380.50 KB (389,632 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msimtf.dll
msctf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 617.50 KB (632,320 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\msctf.dll
jscript 5.6.0.8827 974.50 KB (997,888 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\jscript.dll
imm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 208.00 KB (212,992 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\imm32.dll
mshtmled 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 905.50 KB (927,232 bytes)
3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\mshtmled.dll
vbscript 5.6.0.8827 646.50 KB (662,016 bytes) 3/25/2005 6:00 AM Microsoft Corporation c:\windows\system32\vbscript.dll
msinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 636.00 KB (651,264 bytes)
2/23/2007 12:47 PM Microsoft Corporation
c:\windows\pchealth\helpctr\binaries\msinfo.dll
mfc42u 6.50.9146.0 1.39 MB (1,462,272 bytes) 3/25/2005 6:00 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 6:00 AM Microsoft Corporation c:\windows\system32\riched32.dll
riched20 5.31.23.1224 1.10 MB (1,157,120 bytes) 3/25/2005 6:00 AM
Microsoft Corporation c:\windows\system32\riched20.dll

helpsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.52 MB (1,591,296 bytes)
2/23/2007 12:47 PM 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 AeLookupSvc Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Alerter Alerter Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k localservice Normal NT AUTHORITY\LocalService 0
Application Layer Gateway Service ALG Stopped Manual Own Process
c:\windows\system32\alg.exe Normal NT AUTHORITY\LocalService 0
Application Management AppMgmt Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
ASP.NET State Service aspnet_state Stopped Manual Own Process
c:\windows\microsoft.net\framework64\v2.0.50727\aspnet_state.exe Normal NT AUTHORITY\NetworkService 0
Windows Audio AudioSrv Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Background Intelligent Transfer Service BITS Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Computer Browser Browser Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Indexing Service CiSvc Stopped Disabled Share Process
c:\windows\system32\cisvc.exe Normal LocalSystem 0
ClipBook ClipSrv Stopped Disabled Own Process
c:\windows\system32\clipsrv.exe Normal LocalSystem 0
.NET Runtime Optimization Service v2.0.50727_X86 clr_optimization_v2.0.50727_32
Stopped Manual Own Process
c:\windows\microsoft.net\framework\v2.0.50727\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\mscorsvw.exeIgnore LocalSystem 0
COM+ System Application COMSysApp Stopped Manual Own Process
c:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}
Normal LocalSystem 0
Cryptographic Services CryptSvc Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
DCOM Server Process Launcher DcomLaunch Running Auto Share Process
c:\windows\system32\svchost.exe -k dcomlaunch Normal LocalSystem 0
Distributed File System Dfs Stopped Manual Own Process
c:\windows\system32\dfssvc.exe Normal LocalSystem 0
DHCP Client Dhcp Running Auto Share Process
c:\windows\system32\svchost.exe -k networkservice Normal NT AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service dmadmin Stopped Manual Share Process c:\windows\system32\dmadmin.exe /com Normal LocalSystem 0
Logical Disk Manager dmserver Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0

DNS Client Dnscache Running Auto Share Process
c:\windows\system32\svchost.exe -k networkservice Normal NT
AUTHORITY\NetworkService 0
Error Reporting Service ERSvc Running Auto Share Process
c:\windows\system32\svchost.exe -k winerr Ignore LocalSystem 0
Event Log Eventlog Running Auto Share Process
c:\windows\system32\services.exe Normal LocalSystem 0
COM+ Event System EventSystem Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Help and Support helpsvc Running Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Human Interface Device Access HidServ Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
HTTP SSL HTTPFilter Stopped Manual Share Process
c:\windows\system32\lsass.exe Normal LocalSystem 0
IAS Jet Database Access IASJet Stopped Manual Share Process
c:\windows\syswow64\svchost.exe -k iasjet Normal LocalSystem 0
IMAPI CD-Burning COM Service ImapiService Stopped Disabled Own
Process c:\windows\system32\imapi.exe Normal LocalSystem 0
Intersite Messaging IsmServ Stopped Disabled Own Process
c:\windows\system32\ismserv.exe Normal LocalSystem 0
Kerberos Key Distribution Center kdc Stopped Disabled Share Process
c:\windows\system32\lsass.exe Normal LocalSystem 0
Server lanmanserver Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Workstation lanmanworkstation Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
License Logging LicenseService Stopped Disabled Own Process
c:\windows\system32\llssrv.exe Normal NT AUTHORITY\NetworkService 0
TCP/IP NetBIOS Helper LmHosts Running Auto Share Process
c:\windows\system32\svchost.exe -k localservice Normal NT
AUTHORITY\LocalService 0
Messenger Messenger Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
NetMeeting Remote Desktop Sharing mnmsrvc Stopped Disabled Own
Process c:\windows\system32\mnmsrvc.exe Normal LocalSystem 0
Distributed Transaction Coordinator MSDTC Running Auto Own Process
c:\windows\system32\msdtc.exe Normal NT AUTHORITY\NetworkService 0
SQL Server Integration Services MsDtsServer Stopped Manual Own Process
"c:\program files\microsoft sql server\90\dts\binn\msdtssrvr.exe" Normal NT
AUTHORITY\NetworkService 0
SQL Server FullText Search (MSSQLSERVER) msftesql Stopped Manual Own
Process "c:\program files\microsoft sql server\mssql.1\mssql\binn\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
SQL Server (MSSQLSERVER) MSSQLSERVER Stopped Manual Own Process
"c:\program files\microsoft sql server\mssql.1\mssql\binn\sqlservr.exe" -smssqlserver
Normal LocalSystem 0
SQL Server Active Directory Helper MSSQLServerADHelper Stopped Disabled
Own Process "c:\program files\microsoft sql server\90\shared\sqladhlp90.exe" Normal
NT AUTHORITY\NetworkService 0
SQL Server Analysis Services (MSSQLSERVER) MSSQLServerOLAPService
Stopped Manual Own Process "c:\program files\microsoft sql

server\mssql.2\olap\bin\msmdsrv.exe" -s "c:\program files\microsoft sql server\mssql.2\olap\config" Normal LocalSystem 0
Network DDE NetDDE Stopped Disabled Share Process
c:\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 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 (x86)\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) RpcSs Running Auto Share Process
c:\windows\system32\svchost.exe -k rpcss Normal NT AUTHORITY\NetworkService 0
Resultant Set of Policy Provider RSoPProv Stopped Manual Share Process
c:\windows\system32\rsopprov.exe Normal LocalSystem 0
Special Administration Console Helper sacsvr Running Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Security Accounts Manager SamSs Running Auto Share Process
c:\windows\system32\lsass.exe Normal LocalSystem 0
Smart Card SCardSvr Stopped Manual Share Process
c:\windows\system32\scardsvr.exe Ignore NT AUTHORITY\LocalService 0
Task Scheduler Schedule Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Secondary Logon seclogon Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Ignore LocalSystem 0

System Event Notification SENS Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Windows Firewall/Internet Connection Sharing (ICS) SharedAccess Stopped
Disabled Share Process c:\windows\system32\svchost.exe -k netsvcs Normal
LocalSystem 0
Shell Hardware Detection ShellHWDetection Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Ignore LocalSystem 0
Print Spooler Spooler Stopped Manual Own Process
c:\windows\system32\spoolsv.exe Normal LocalSystem 0
SQL Server Browser SQLBrowser Stopped Disabled Own Process
"c:\program files (x86)\microsoft sql server\90\shared\sqlbrowser.exe" Normal
LocalSystem 0
SQL Server Agent (MSSQLSERVER) SQLSERVERAGENT Stopped Manual Own
Process "c:\program files\microsoft sql server\mssql.1\mssql\binn\sqlagent90.exe" -i
mssqlserver Normal LocalSystem 0
SQL Server VSS Writer SQLWriter Stopped Manual Own Process "c:\program
files\microsoft sql server\90\shared\sqlwriter.exe" Normal LocalSystem 0
Windows Image Acquisition (WIA) stisvc Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k imgsvc Normal NT AUTHORITY\LocalService
0
Microsoft Software Shadow Copy Provider swprv Stopped Manual Own Process
c:\windows\system32\svchost.exe -k swprv Normal LocalSystem 0
Performance Logs and Alerts SysmonLog Stopped Auto Own Process
c:\windows\system32\smlogsvc.exe Normal NT Authority\NetworkService 0
Telephony TapiSrv Stopped Manual Share Process
c:\windows\system32\svchost.exe -k tapisrv Normal LocalSystem 0
Terminal Services TermService Running Manual Share Process
c:\windows\system32\svchost.exe -k termsvcs Normal LocalSystem 0
Themes Themes Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Telnet TlntSvr Stopped Disabled Own Process c:\windows\system32\tlntsvr.exe
Normal NT AUTHORITY\LocalService 0
Distributed Link Tracking Server TrkSvr Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Distributed Link Tracking Client TrkWks Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Terminal Services Session Directory Tssdis Stopped Disabled Own Process
c:\windows\system32\tssdis.exe Normal LocalSystem 0
Windows User Mode Driver Framework UMWdf Stopped Manual Own Process
c:\windows\system32\wdfmgr.exe Normal NT AUTHORITY\LocalService 0
Uninterruptible Power Supply UPS Stopped Manual Own Process
c:\windows\system32\ups.exe Normal 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 Running Auto Share Process
c:\windows\system32\svchost.exe -k localservice Normal NT
AUTHORITY\LocalService 0
WebClient WebClient Stopped Disabled Share Process
c:\windows\system32\svchost.exe -k localservice Normal NT
AUTHORITY\LocalService 0
WinHTTP Web Proxy Auto-Discovery Service WinHttpAutoProxySvc Stopped Manual
Share Process c:\windows\system32\svchost.exe -k localservice Normal NT
AUTHORITY\LocalService 0

Windows Management Instrumentation winmgmt Running Auto Share Process
c:\windows\system32\svchost.exe -k netsvcs Ignore LocalSystem 0
Portable Media Serial Number Service WmdmPmSN Stopped Manual Share Process
c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0
Windows Management Instrumentation Driver Extensions Wmi Stopped Manual
Share Process c:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem
0
WMI Performance Adapter WmiApSrv Stopped Manual Own Process
c:\windows\system32\wbem\wmiapsrv.exe Normal LocalSystem 0
Automatic Updates wuauserv 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 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
Broadcom All Users:Broadcom All Users
Microsoft SQL Server 2005 All Users:Microsoft SQL Server 2005 All Users
Microsoft SQL Server 2005\Configuration Tools All Users:Microsoft SQL Server
2005\Configuration Tools All Users
Startup All Users:Startup All Users
Accessories NT AUTHORITY\SYSTEM:Accessories NT AUTHORITY\SYSTEM
Accessories\Accessibility NT AUTHORITY\SYSTEM:Accessories\Accessibility NT
AUTHORITY\SYSTEM
Accessories\Entertainment NT AUTHORITY\SYSTEM:Accessories\Entertainment NT
AUTHORITY\SYSTEM
Startup NT AUTHORITY\SYSTEM:Startup NT AUTHORITY\SYSTEM
Accessories PE2900\Administrator:Accessories PE2900\Administrator
Accessories\Accessibility PE2900\Administrator:Accessories\Accessibility
PE2900\Administrator
Accessories\Entertainment PE2900\Administrator:Accessories\Entertainment
PE2900\Administrator
Startup PE2900\Administrator:Startup PE2900\Administrator

[Startup Programs]

Program Command User Name Location
desktop desktop.ini NT AUTHORITY\SYSTEM Startup
desktop desktop.ini PE2900\Administrator Startup
desktop desktop.ini .DEFAULT Startup
desktop desktop.ini All Users Common Startup

bacstray c:\program files\broadcom\bacs\bacstray.exe All Users
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

[OLE Registration]

Object Local Server
Sound (OLE2) sndrec32.exe
Media Clip mplay32.exe
Video Clip mplay32.exe /avi
MIDI Sequence mplay32.exe /mid
Sound Not Available
Media Clip Not Available
WordPad Document "%programfiles%\windows nt\accessories\wordpad.exe"
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 Company
actxprxy.dll 6.0.3790.1830 221 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
actxprxy.dll 6.0.3790.1830 221 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
advpack.dll 6.0.3790.1830 146 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
advpack.dll 6.0.3790.1830 146 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
asctrls.ocx 6.0.3790.1830 147 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
asctrls.ocx 6.0.3790.1830 147 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
browselc.dll 6.0.3790.1830 63 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
browselc.dll 6.0.3790.1830 63 KB 3/25/2005 6:00:00 AM . Microsoft Corporation

browseui.dll 6.0.3790.1830 1,564 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
browseui.dll 6.0.3790.1830 1,564 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation
cdfview.dll 6.0.3790.1830 216 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
cdfview.dll 6.0.3790.1830 216 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
comctl32.dll 5.82.3790.1830 935 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
comctl32.dll 5.82.3790.1830 935 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
dxtrans.dll 6.3.3790.1830 320 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
dxtrans.dll 6.3.3790.1830 320 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
dxtmsft.dll 6.3.3790.1830 549 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
dxtmsft.dll 6.3.3790.1830 549 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
iecont.dll <File Missing> Not Available Not Available Not Available Not Available
iecontlc.dll <File Missing> Not Available Not Available Not Available Not Available
iedkcs32.dll 16.0.3790.1830 417 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
iedkcs32.dll 16.0.3790.1830 417 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
iepeers.dll 6.0.3790.1830 361 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
iepeers.dll 6.0.3790.1830 361 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
iesetup.dll 6.0.3790.1830 71 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
iesetup.dll 6.0.3790.1830 71 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
ieuinit.inf Not Available 24 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Not Available
ieuinit.inf Not Available 24 KB 3/25/2005 6:00:00 AM . Not Available
iexplore.exe 6.0.3790.1830 94 KB 3/25/2005 6:00:00 AM C:\Program Files\Internet
Explorer Microsoft Corporation
imgutil.dll 6.0.3790.1830 61 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
imgutil.dll 6.0.3790.1830 61 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
inetcpl.cpl 6.0.3790.1830 428 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
inetcpl.cpl 6.0.3790.1830 428 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
inetcplc.dll 6.0.3790.1830 110 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
inetcplc.dll 6.0.3790.1830 110 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
inseng.dll 6.0.3790.1830 147 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
inseng.dll 6.0.3790.1830 147 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
mlang.dll 6.0.3790.1830 686 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
mlang.dll 6.0.3790.1830 686 KB 3/25/2005 6:00:00 AM . 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 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
mshta.exe 6.0.3790.1830 38 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
mshtml.dll 6.0.3790.1830 5,790 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.dll 6.0.3790.1830 5,790 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation

mshtml.tlb 6.0.3790.1830 1,320 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
mshtml.tlb 6.0.3790.1830 1,320 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation
mshtmled.dll 6.0.3790.1830 906 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
mshtmled.dll 6.0.3790.1830 906 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
mshtmler.dll 6.0.3790.1830 56 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
mshtmler.dll 6.0.3790.1830 56 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
msident.dll 6.0.3790.1830 69 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
msident.dll 6.0.3790.1830 69 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
msidntld.dll 6.0.3790.1830 16 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
msidntld.dll 6.0.3790.1830 16 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
msieftp.dll 6.0.3790.1830 369 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
msieftp.dll 6.0.3790.1830 369 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
msrating.dll 6.0.3790.1830 240 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
msrating.dll 6.0.3790.1830 240 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
mstime.dll 6.0.3790.1830 878 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
mstime.dll 6.0.3790.1830 878 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
occache.dll 6.0.3790.1830 126 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
occache.dll 6.0.3790.1830 126 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
proctexe.ocx <File Missing> Not Available Not Available Not Available Not Available
sendmail.dll 6.0.3790.1830 64 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
sendmail.dll 6.0.3790.1830 64 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
shdoclc.dll 6.0.3790.1830 590 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
shdoclc.dll 6.0.3790.1830 590 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
shdocvw.dll 6.0.3790.1830 2,360 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
shdocvw.dll 6.0.3790.1830 2,360 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation
shfolder.dll 6.0.3790.1830 34 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
shfolder.dll 6.0.3790.1830 34 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
shlwapi.dll 6.0.3790.1830 607 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
shlwapi.dll 6.0.3790.1830 607 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
tdc.ocx 1.3.0.3130 91 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
tdc.ocx 1.3.0.3130 91 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
url.dll 6.0.3790.1830 40 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
url.dll 6.0.3790.1830 40 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
urlmon.dll 6.0.3790.1830 1,049 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
urlmon.dll 6.0.3790.1830 1,049 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation

webcheck.dll 6.0.3790.1830 439 KB 3/25/2005 6:00:00 AM C:\WINDOWS\system32
Microsoft Corporation
webcheck.dll 6.0.3790.1830 439 KB 3/25/2005 6:00:00 AM . Microsoft Corporation
wininet.dll 6.0.3790.1830 1,159 KB 3/25/2005 6:00:00 AM
C:\WINDOWS\system32 Microsoft Corporation
wininet.dll 6.0.3790.1830 1,159 KB 3/25/2005 6:00:00 AM . Microsoft
Corporation

[Connectivity]

Item Value
Connection Preference Never dial

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Enabled
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 High
Restricted sites Custom

RTE Input Parameters

Profile: 5512_pe2900
File Path: C:\Program Files\BenchCraft\5512_pe2900.xml
Version: 5

Number of Engines: 6

Name: DRIVER3
Description: rte104_1
Directory: c:\tpcclog\rte104_1.log
Machine: rte104
Parameter Set: PARAM2
Index: 200000000
Seed: 59915
Configured Users: 9550
Pipe Name: DRIVER3-1689047983
Connect Rate: 160
Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: callnewordernew=1

Name: DRIVER4
Description: rte104_2
Directory: c:\tpcclog\rte104_2.log
Machine: rte104
Parameter Set: PARAM2
Index: 300000000
Seed: 59915
Configured Users: 9550
Pipe Name: DRIVER4190963968
Connect Rate: 160

Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options: callnewordernew=1

Name: DRIVER1
Description: rte103_1
Directory: c:\tpcclog\rte103_1.log
Machine: rte103
Parameter Set: PARAM2
Index: 200000000
Seed: 59915
Configured Users: 9550
Pipe Name: DRIVER3148536062
Connect Rate: 160
Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: callnewordernew=1

Name: DRIVER2
Description: rte103_2
Directory: c:\tpcclog\rte103_2.log
Machine: rte103
Parameter Set: PARAM2
Index: 300000000
Seed: 59915
Configured Users: 9550
Pipe Name: DRIVER4148635359
Connect Rate: 160
Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: callnewordernew=1

Name: DRIVER5
Description: rte103_3
Directory: c:\tpcclog\rte103_3.log
Machine: rte103
Parameter Set: PARAM2
Index: 400000000
Seed: 59915
Configured Users: 7250
Pipe Name: DRIVER5162429328
Connect Rate: 160
Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233

CPU: 0
Additional Options: callnewordernew=1

Name: DRIVER6
Description: rte104_3
Directory: c:\tpcclog\rte104_4.log
Machine: rte104
Parameter Set: PARAM2
Index: 500000000
Seed: 59915
Configured Users: 9670
Pipe Name: DRIVER6162485953
Connect Rate: 160
Start Rate: 160
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options: callnewordernew=1

Number of User groups: 6

Driver Engine: DRIVER3
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 955
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 9550
District id: 1
Scale Down: No

Driver Engine: DRIVER4
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 956 - 1910
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 9550
District id: 1
Scale Down: No

Driver Engine: DRIVER1
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa

Protocol: HTML
w_id Range: 1911 - 2865
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 9550
District id: 1
Scale Down: No

Driver Engine: DRIVER2
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2866 - 3820
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 9550
District id: 1
Scale Down: No

Driver Engine: DRIVER5
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3821 - 4545
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 7250
District id: 1
Scale Down: No

Driver Engine: DRIVER6
IIS Server: client1430
SQL Server: pe2900
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 4546 - 5512
w_id Min Warehouse: 1
w_id Max Warehouse: 5512
Scale: Normal
User Count: 9670
District id: 1
Scale Down: No

Number of Parameter Sets: 5

~Default

Default Parameter Set

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay	
New Order	10.00		12.05	18.01	0.10	5.00	0.10
Payment	10.00		12.05	3.01	0.10	5.00	0.10
Delivery	1.00		5.05	2.01	0.10	5.00	0.10
Stock Level	1.00		5.05	2.01	0.10	20.00	0.10
Order Status	1.00		10.05	2.01	0.10	5.00	0.10

PARAM2

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay	
New Order	44.84		12.04	18.02	0.10	5.00	0.10
Payment	43.04		12.04	3.02	0.10	5.00	0.10
Delivery	4.05		5.04	2.02	0.10	5.00	0.10
Stock Level	4.05		5.04	2.02	0.10	20.00	0.10
Order Status	4.05		10.04	2.02	0.10	5.00	0.10

50run

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay	
New Order	44.84		30.00	18.02	0.10	5.00	0.10
Payment	43.04		30.00	3.02	0.10	5.00	0.10
Delivery	4.05		15.00	2.02	0.10	5.00	0.10
Stock Level	4.05		15.00	2.02	0.10	20.00	0.10
Order Status	4.05		25.00	2.02	0.10	5.00	0.10

50run2

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay	
New Order	44.84		33.00	18.02	0.10	5.00	0.10
Payment	43.04		33.00	3.02	0.10	5.00	0.10
Delivery	4.05		18.00	2.02	0.10	5.00	0.10
Stock Level	4.05		18.00	2.02	0.10	20.00	0.10
Order Status	4.05		28.00	2.02	0.10	5.00	0.10

80run

	Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay	
New Order	44.84		19.00	18.02	0.10	5.00	0.10
Payment	43.04		19.00	3.02	0.10	5.00	0.10
Delivery	4.05		14.00	2.02	0.10	5.00	0.10
Stock Level	4.05		14.00	2.02	0.10	20.00	0.10
Order Status	4.05		9.00	2.02	0.10	5.00	0.10

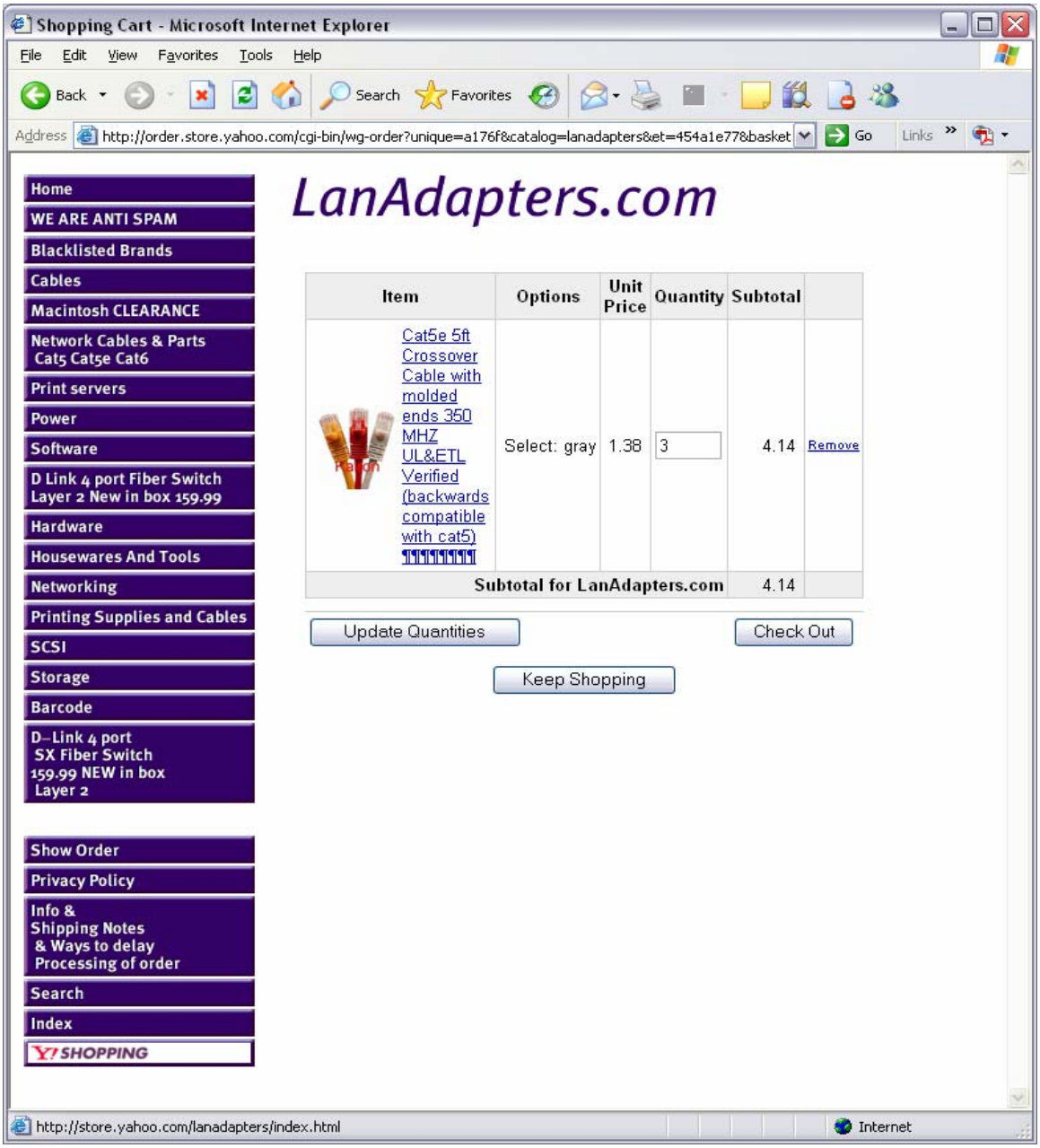
Appendix D – Disk Storage

Number of Warehouses:	5,512
Warehouse Table	
Cardinality:	5,512
Data Space:	0.60 MB
Data Space (New):	0.54 MB
Clustered Index Space:	0.02 MB
Clustered Index Space (New):	0.01 MB
Non-Clustered Index Space:	0.02 MB
Table Size (Data + Index):	1.19 MB
Table Size (Data + Index) (New):	0.57 MB
Total Space Required (Table Size + 5%):	0.67 MB
District Table	
Cardinality:	55,120
Data Space:	6.28 MB
Data Space (New):	5.91 MB
Clustered Index Space:	0.22 MB
Clustered Index Space (New):	0.02 MB
Non-Clustered Index Space:	0.22 MB
Table Size (Data + Index):	12.65 MB
Table Size (Data + Index) (New):	6.15 MB
Total Space Required (Table Size + 5%):	7.04 MB
Customer Table	
Cardinality:	165,360,000
Data Space:	113,602.32 MB
Data Space (New):	117,444.00 MB
Clustered Index Space:	662.75 MB
Clustered Index Space (New):	334.61 MB
Non-Clustered Index Space:	9,033.28 MB
Non-Clustered Index Space (New):	10,930.46 MB
Table Size (Data + Index):	241,076.97 MB
Table Size (Data + Index) (New):	128,709.07 MB
Total Space Required (Table Size + 5%):	128,978.47 MB
History Table	
Cardinality:	165,360,000
Data Space:	9,590.88 MB
Clustered Index Space:	661.44 MB
Non-Clustered Index Space:	661.44 MB
Table Size (Data + Index):	10,913.76 MB
Total Space Required (Table Size + 5%):	11,393.30 MB
Orders Table	
Cardinality:	165,360,000
Data Space:	5,456.88 MB
Clustered Index Space:	662.75 MB
Non-Clustered Index Space:	4,311.16 MB
Table Size (Data + Index):	10,430.79 MB
Total Space Required (Table Size + 5%):	10,703.64 MB
New Order Table	
Cardinality:	49,608,000
Data Space:	892.94 MB
Clustered Index Space:	198.83 MB
Non-Clustered Index Space:	198.43 MB
Table Size (Data + Index):	1,290.20 MB
Total Space Required (Table Size + 5%):	1,334.85 MB
Order Line Table	
Cardinality:	1,653,600,000
Data Space:	107,484.00 MB
Clustered Index Space:	6,628.38 MB
Non-Clustered Index Space:	6,614.40 MB
Table Size (Data + Index):	120,726.78 MB
Total Space Required (Table Size + 5%):	126,100.98 MB
Stock Table	
Cardinality:	551,200,000
Data Space:	175,832.80 MB
Clustered Index Space:	2,208.91 MB
Non-Clustered Index Space:	2,204.80 MB
Table Size (Data + Index):	180,246.51 MB
Total Space Required (Table Size + 5%):	189,038.15 MB
Item Table	
Cardinality:	100,000
Data Space:	9.80 MB
Clustered Index Space:	0.40 MB
Non-Clustered Index Space:	0.40 MB
Table Size (Data + Index):	10.60 MB
Total Space Required (Table Size + 5%):	11.09 MB
Minimum Space Required for Load and Index:	467,568.19 MB
Number of Users:	55,120
Minimum Throughput (9 tpmC per warehouse):	49,608.00 tpmC
Maximum Throughput (12.86 tpmC per warehouse):	70,884.32 tpmC
Average Throughput (12.28 tpmC per warehouse):	67,687.36 tpmC
Average initial space required per tpmC:	6.91 MB

Steady State Period (minutes):	960
Average Transaction Counts	
New-Order Trans executed during steady state:	64,979,866
Payment Trans executed during steady state (43%):	64,979,866
Order Status Trans executed during steady state (4%):	5,907,261
Stock Level Trans executed during steady state (4%):	5,907,261
Delivery Trans executed during steady state (4%):	5,907,261
Average Dynamic Table Growth:	
Growth in order table during steady state (KB):	6,978,188
Growth in order_line table during steady state (rows):	41,637,798
Growth in history table during steady state (rows):	3,978,067
Growth in new-order table during steady state (rows):	712,829
Growth in order table size during steady state (MB):	6,814.64 MB
Growth in order-line table size during steady state (MB):	40,661.91 MB
Growth in history table size during steady state (MB):	3,884.83 MB
Growth in new-order table size during steady state (MB):	696.12 MB
Growth of all Dynamic Tables during steady state (MB):	52,057.50 MB
Initial Database Size:	467,568.19 MB
Estimated Database Size after Steady State:	519,625.70 MB
% Increase Database size after Steady State:	11.13%
Avg. Log Space per tpmC:	5.42 KB
Log Space Required for TPC-C Steady State:	343,937.00 MB

	43636368
	42,614.00 MB
	2722752
	2,659.00 MB
	45,273.00 MB
	114,265.07 MB
	124777.4619

Appendix E - Price Quotations



Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

March 2, 2007

Dell
Dan Hambrick
1 Dell Way
Round Rock, TX 78664

Mr. Hambrick:

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
228-04026	**SQL Server 2005 Standard x64 Edition** *Per Processor License* *Discount Schedule: No Discounts Applied*	$5,999	1	$5,999
P73-0295	**Windows Server 2003, Standard x64 Edition** *Server License Only - No CALs* *Discount Schedule: Open Program – No Level* *Unit Price reflects a 28% discount from the retail unit price of $999.*	$719	1	$719
P73-00295	**Windows Server 2003 Standard Edition** *Server License Only - No CALs* *Discount Schedule: Open Program – No Level* *Unit Price reflects a 28% discount from the retail unit price of $999.*	$719	1	$719
254-00170	**Visual C++ Standard Edition** *No Discounts Applied*	$109	1	$109
N/A	**Microsoft Problem Resolution Services** *Professional Support* *(1 Incident)*	$245	1	$245

All products are currently orderable through Microsoft's normal distribution channels. A list of Microsoft's authorized resellers can be found at: http://www.microsoft.com/products/info/render.aspx?view=22&type=mnp&content=22/licensing.

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.