Compaq Computer Corporation

TPC Benchmark™ C
Full Disclosure Report
for
ProLiant ML570-3P
using
Microsoft SQL Server 2000 Standard Edition
and
Windows 2000 Server

First Edition August 2000
Second Edition July 2001
Upgraded to TPC-C Version 5.0 July 27, 2001

COMPAQ

Second Edition – July 2001

Compaq Computer Corporation (Compaq) believes that the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. Compaq assumes no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, Compaq provides no warranty of the pricing information in this document.

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

All performance data contained in this report were obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. Compaq does not warrant or represent that a user can or will achieve similar performance expressed in transactions per minute (tpmC) or normalized price/performance ($/tpmC). No warranty of system performance or price/performance is expressed or implied in this report.

Printed in U.S.A., 2001
Compaq, NonStop, ProLiant ML570, and ProLiant are registered trademarks of Compaq Computer Corporation.

Microsoft, Windows 2000 and SQL Server 2000 are registered trademarks of Microsoft Corporation.

Pentium III Xeon is a registered trademark of Intel.

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

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

Table of Contents

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 3.5, released August 26, 1999. It is also upgraded per TPC rules to TPC-C Version 5.0.

TPC Benchmark C Overview

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

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

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

The performance metric reported by TPC-C is a "business throughput" measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments are not recommended.

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

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted on the Compaq ProLiant ML570. The operating system used for the benchmark was Windows 2000 Server. The DBMS used was Microsoft SQL Server 2000 Standard Edition.

TPC Benchmark C Metrics

The standard TPC Benchmark C metrics, tpmC (transactions per minute), price per tpmC (five year capital cost per measured tpmC), and the availability date are reported as:

20207.20 tpmC
$5.64 per tpmC

The availability date is September 26, 2000.

Standard and Executive Summary Statements

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

Auditor

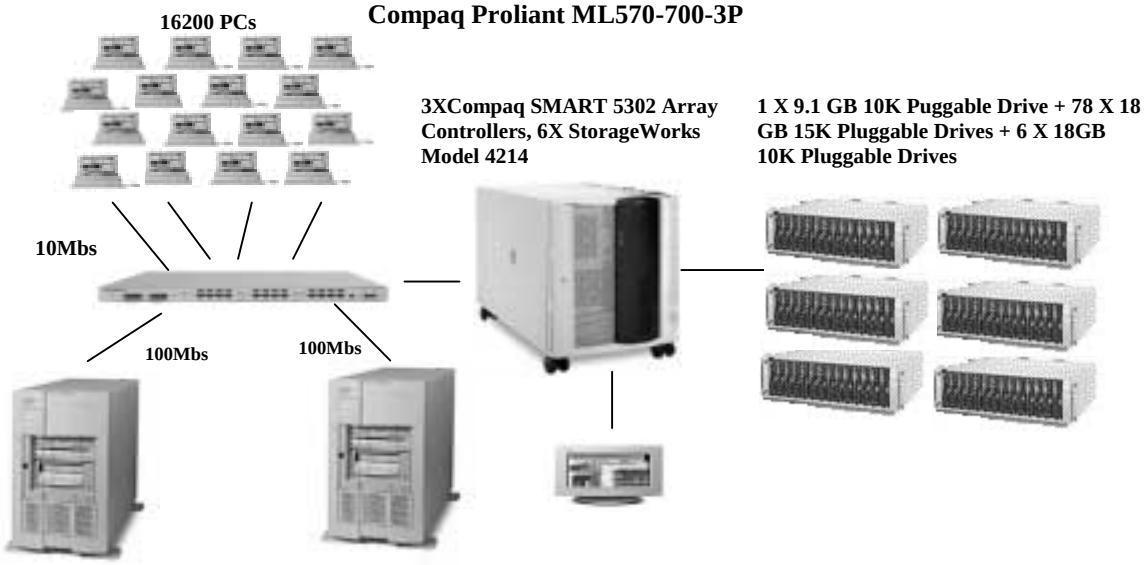
The benchmark configuration, environment and methodology were audited by Lorna Livingtree of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

Compaq Computer Corporation	ProLiant ML570-700-3P Upgrade Date July 27, 2001 C/S with 2 ProLiant ML350	TPC-C Rev. 5.0 Upgrade Report Date: Aug 21,2000

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$113,798	**20207.20**	**$5.64**	**Sept 26, 2000**

Processors	Database	Operating System	Other Software	Number Users
3 Pentium III Xeon 700 MHz	Microsoft SQL Server 2000 Standard Edition	Windows 2000 Server	Microsoft Visual C++ Microsoft COM+	**16200**

Compaq Proliant ML570-700-3P



	Server		Each Client	
System Components	Quantity	Description	Quantity	Description
Processor	2	700MHz Pentium III Xeon w/ 1MB Cache	1	Pentium III 933MHz , 256KB cache 10/100 T/X NIC,
	1	700MHz Pentium III Xeon w/ 2MB Cache		
Memory	8	256MB	1	128MB
	4	128MB	1	256MB
Disk Controllers	3	SMART SCSI Array Controller 5302	1	Integrated Dual Wide-Ultra2 controller,
	1	Integrated SCSI Array Controller		
Disk Drives	1	9.1GB SCSI Drive	1	9.1GB SCSI Drive
	84	18GB SCSI Drive		
Total Storage		1.4274 TeraBytes		9.1 GB
Tape Drives	1	12/24-GB DAT Drive		

Compaq Computer Corporation	ProLiant ML570-700	TPC-C Rev. 5.0	
	Client/Server	Report Date:	27-Jul-01

Description	Part Number	Third Party Brand	Unit Price	Qty	Extended Price	3 yr. Maint. Price
Server Hardware			Pricing			
Proliant ML570T X700 1MB 512	138665-001		1 6,307	1	6,307	
- 1 Pentium III Xeon/700MHz 1MB Cache						
- 512MB SDRAM Memory						
- CD-ROM 24X ,NC3134 10/100 NIC						
XN700 1M S2 Processor Kit	174448-B21		1 1,759	1	1,759	
XN700 2M S2 Processor Kit	174449-B21		1 3,099	1	3,099	
1GB SDRAM (4x256) memory module	189081-B21		1 1,099	2	2,198	
Integrated Smart Array Controller	128293-B21		1 454	1	454	
StorageWorks Enclosure Model 4214R	190209-001		1 2,913	6	17,478	
Compaq SMART Array Controller 5302 - 2 SCSI Chan.	166207-B21		1 1,299	3	3,897	
Compaq V570 Color Monitor	228114-001		1 179	1	179	
12/24-Gigabyte DAT Drive (Internal)	295513-B22		1 682	1	682	
9.1-GB Pluggable Wide Ultra2 Universal 10K Drive (1")	328939-B22		1 477	1	477	
18.2 GB Hot-Plug Wide U3 15K 1"	188122-B22		1 670	78	52,260	
18.2GB Pluggable Ultra2 SCSI Universal 10K Drive (1")	142673-B22		1 423	6	2,538	
18.2 GB Hot-Plug Wide U3 15K 1" (10% spares for all driv	188122-B22		1 670	9		6,030
CarePaq Service - Departmental Servers 3Yr,7x24,4hr Re	FM-MI724-36		1 1,795	1		1,795
CarePaq Service - 42xx/43xx Enclosure 3Yr,7x24,4hr Res	FM-4E724-36		1 157	6		942
				Subtotal	**91,328**	**8,767**
Server Software						
Microsoft SQL Server 2000 Standard (per processor)		Microsoft	2 12,813	1	12,813	6,285
Microsoft Visual C++ 6.0		Microsoft	2 549	1	549	Incl Above
Microsoft Windows 2000 Server		Microsoft	2 738	1	738	Incl Above
				Subtotal	**14,100**	**6,285**
Client Hardware						
ProLiant ML350T P933/133 128MB NHP	170475-001		1 1,449	2	2,898	
-- Pentium III 933MHz , 256KB level two ECC cache, -------- -- Integrated Dual Wide-Ultra2 controller, 10/100 T/X NIC,						
256 Reg 133MHz SDRAM DIMM	128278-B21		1 250	2	500	
NC3123 Fast Ethernet NIC PCI 10/100 Wake on LAN	174830-B21		1 98	2	196	
Compaq V570 Color Monitor	228114-001		1 179	2	358	
9.1 Gigabyte Wide Ultra2 SCSI Hard Drive	120204-B21		1 434	2	868	
CarePaq Service - Entry Workgroup Servers 3Yr,7x24,4hr	FM-EL724-36		1 750	2		1,500
				Subtotal	**4,820**	**1,500**
Client Software						
Microsoft Windows 2000 Server		Microsoft	2 738	2	1,476	Incl. Above
				Subtotal	**1,476**	**0**
Connectivity						
LinkSys 8 Port 10/100 Switch EZX#88R	DEH4162	Linksys	3 140	3	**420**	**See Note 1**
Large Purchase and Cash discount (See Note 2)	14.0%		1		**($13,461)**	**($1,437)**
				Total	**$98,683**	**$15,115**

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

Five-Year Cost of Ownership: $113,798

tpmC Rating: 20207.20

$ / tpmC: $5.64

Pricing: 1=Compaq Direct 2= Microsoft 3=PC Connection Note 1 = 5 Year warranty with 10% Spares - Note 2 = Discount based on Compaq Direct guidence and large cash purchase level.

Note:The benchmark results and test methodology were audited by Lorna Livingtree of Performance Metrics, Inc.

Numerical Quantities Summary

MQTH, Computed Maximum Qualified Throughput 20207.20 tpmC

% throughput difference, reported & reproducibility runs 0.398%

Response Times (in seconds)	Average	90%	Maximum
New-Order	0.38	0.54	2.54
Payment	0.21	0.33	2.30
Order-Status	0.26	0.39	2.48
Delivery (interactive portion)	0.10	0.11	0.16
Delivery (deferred portion)	0.77	1.25	4.02
Stock-Level	2.59	3.49	5.53
Menu	0.10	0.11	0.63

Transaction Mix, in percent of total transaction

New-Order	44.83%
Payment	43.04%
Order-Status	4.04%
Delivery	4.05%
Stock-Level	4.04%

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

Keying/Think Times (in seconds)	Min.	Average	Max.
New-Order	18.00/0.00	18.01/12.10	18.03/121.20
Payment	3.00/0.00	3.01/12.11	3.03/121.21
Order-Status	2.00/0.00	2.01/10.09	2.03/100.99
Delivery (interactive)	2.00/0.00	2.01/5.07	2.02/50.49
Stock-Level	2.00/0.00	2.01/5.04	2.02/50.49

Test Duration

Ramp-up time	69 minutes
Measurement interval	20 minutes
Transactions (all types) completed during measurement interval	901466
Ramp down time	54 minutes

Checkpointing

Number of checkpoints	1
Checkpoint interval	20 minutes

General Items

Test Sponsor

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

This benchmark was sponsored by Compaq Computer Corporation. The benchmark was developed and engineered by Compaq Computer Corporation. Testing took place at Compaq benchmarking laboratories in Houston, Texas.

Application Code and Definition Statements

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

Appendix A contains all source code implemented in this benchmark.

Parameter Settings

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

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

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

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

Configuration Items

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

The configuration diagrams for both the tested and priced systems are included on the following pages.

Figure 1. Benchmarked Configuration

Compaq Proliant ML570-700-3P

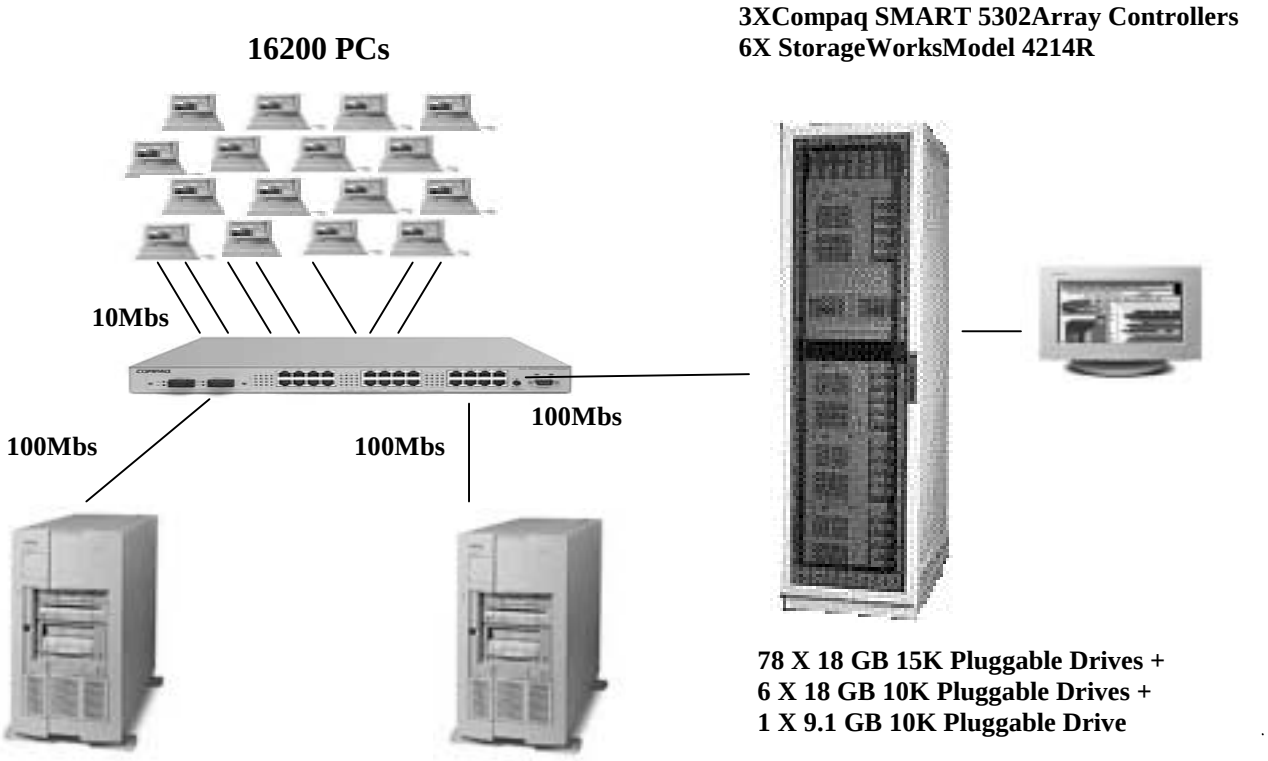
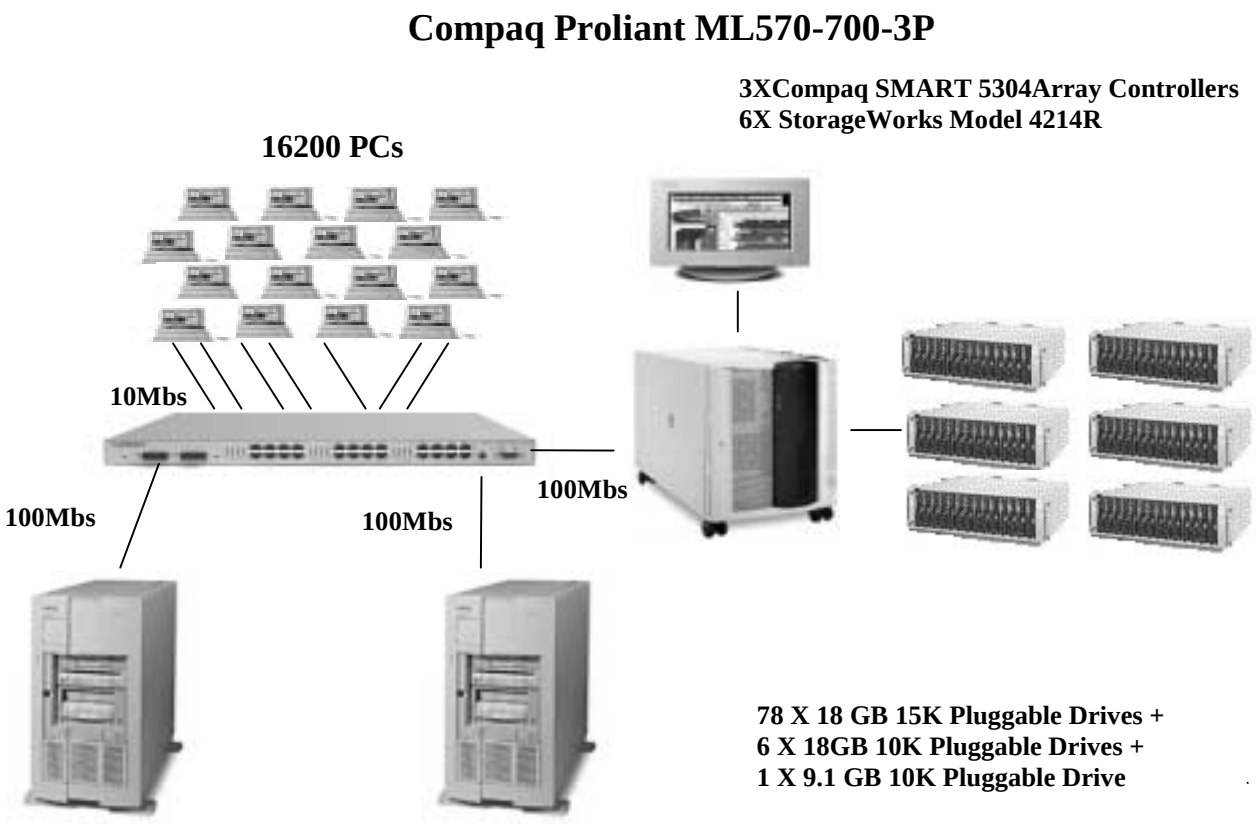


Figure 2. Priced Configuration



Clause 1 Related Items

Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database.

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

Physical Organization of Database

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

The tested configuration consisted of: 84 drives at 18GB, and 1 drive at 9.1GB each.

Benchmarked Configuration:

Embedded Raid Controller

LOGICAL DRIVE C:	Total Capacity = 8.46 GB	RAID 0
Microsoft Windows 2000 Advanced Server		
LOGICAL DRIVE F:	Total Capacity = 50.87 GB	RAID 0+1
MSSQL70_tpcc_log		

SMART-5302 Controller, Slot 2 Array A

LOGICAL DRIVE I:	Total Capacity = 19.53 GB	RAID 0
MSSQL70_misc_fg		
LOGICAL DRIVE M:	Total Capacity = 39.06 GB	RAID 0
MSSQL70_cs_fg		
LOGICAL DRIVE X:	Total Capacity = 191.15 GB	RAID 0+1
Tpccbackup_1		

SMART-5302 Controller, Slot 5, Array A

LOGICAL DRIVE J:	Total Capacity = 19.53 GB	RAID 0
MSSQL70_cs_fg		
LOGICAL DRIVE N:	Total Capacity = 39.06 GB	RAID 0
MSSQL70_misc_fg		
LOGICAL DRIVE Y:	Total Capacity = 191.15 GB	RAID 0+1
Tpccbackup_2		

SMART-5302 Controller, Slot 6, Array A

LOGICAL DRIVE K:	Total Capacity = 19.53 GB	RAID 0
MSSQL70_cs_fg		
LOGICAL DRIVE O:	Total Capacity = 39.06 GB	RAID 0
MSSQL70_misc_fg		
LOGICAL DRIVE Z:	Total Capacity = 191.15 GB	RAID 0+1
Tpccbackup_3		

Priced Configuration vs. Measured Configuration:

The measured configuration has the backup devices stored on 18GB drives, whereas the priced configuration has a tape drive for backup. The measured configuration also used a rack-mounted Proliant ML570 whereas the priced configuration has a tower model.

Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the minimum key value for these new rows.

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

Partitioning

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

No partitioning was used in this benchmark.

Replication, Duplication or Additions

Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used in this benchmark.

Clause 2 Related Items

Random Number Generation

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

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

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

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

Input/Output Screen Layout

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

All screen layouts followed the specifications exactly.

Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal attributes were verified by the auditor manually exercising each specification on a representative Compaq ProLiant ML350 server.

Presentation Manager or Intelligent Terminal

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

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

Transaction Statistics

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

Table 2. 1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.01%
	Remote warehouse order lines	0.99%
	Rolled back transactions	0.98%
	Average items per order	10.00
Payment	Home warehouse payments	85.06%
	Remote warehouse payments	14.94%
	Accessed by last name	60.03%

Statistic		Value
Order Status	Accessed by last name	59.84%
Transaction Mix	New Order	44.83%
	Payment	43.04%
	Order status	4.04%
	Delivery	4.05%
	Stock level	4.04%

Queuing Mechanism

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

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

The source code is listed in Appendix A.

Clause 3 Related Items

Transaction System Properties (ACID)

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

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

Atomicity

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

Completed Transactions

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

Aborted Transactions

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

Consistency

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

Consistency conditions one through four were tested using a 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 an hour 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 seven were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

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

For Isolation test seven, case A was followed.

Durability

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

Durable Media Failure

Loss of Data and Log

To demonstrate recovery from a permanent failure of durable medium containing DBMS logs and TPC-C tables, the following steps were executed:

- The database was backed up to extra disks.
- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 100 users.
- The test was allowed to run for a minimum of 10 minutes.
- One log disk was removed from the drive cabinet.
- Since the disk was mirrored, processing was not interrupted. This was verified by checking the users status on the RTE.
- One of the data disks was removed from the drive cabinet.
- When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down.
- A dump of the transaction log was taken and the Microsoft SQL Server was shutdown.
- A new log disk was inserted into the log drive cabinet. A new data disk was inserted into the data drive cabinet. Both cabinets were powered back up and system restarted.
- Microsoft SQL Server was started.
- The database was restored from backup and the transaction log dump.
- Consistency condition #3 was executed and verified.
- Step 2 was repeated and the difference between the first and second counts was noted.
- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 13 and 14 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database of 1620 warehouses under a full load of 16200 users. The following steps were executed:

- The total number of New Orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
- The RTE was started with 16200 users.
- The test was allowed to run for a minimum of 10 minutes.
- A checkpoint was performed.
- System crash and loss of memory were induced by switching the power off. No battery backup or Uninterruptible Power Supply (UPS) were used to preserve the contents of memory.
- The RTE was shutdown.
- Power was restored and the system restarted.
- Microsoft SQL Server was restarted and performed an automatic recovery.
- Consistency condition #3 was executed and verified.
- Step 1 was repeated and the difference between the first and second counts was noted.

- An RTE report was generated for the entire run time giving the number of NEW-ORDERS successfully returned to the RTE.
- The counts in step 10 and 11 were compared and the results verified that all committed transactions had been successfully recovered.
- Samples were taken from the RTE files and used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table.

Clause 4 Related Items

Initial Cardinality of Tables

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

Table 4.1 Number of Rows for Server

Table	Cardinality as built
Warehouse	1,620
District	16,200
Customer	48,600,000
History	48,600,000
Orders	48,600,000
New Order	14,580,000
Order Line	485,997,340
Stock	162,000,000
Item	100,000
Deleted Warehouses	0

Database Layout

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

The benchmarked configuration used 3 SMART-5302 Array controllers, each with 2 SCSI channels. Each controller is capable of accessing up to 28 disk drives per array, 14 disk drives per each channel, and supports RAID 0 and RAID 1 per each logical volume configured. The data tables were stored on 3 RAID arrays of 26 18GB 15K drives each. Each of these RAID arrays contained 2 logical drives for database data. Each array was configured with RAID 0 and the Array Accelerator was disabled. On the integrated RAID array controller, a RAID array of 6 18GB 10K drives contained 1 logical volume, was configured as RAID 0+1 and stored the transaction log. There is no write cache on the integrated RAID array controller. The operating system was stored on a 9.1GB drive on the integrated RAID array controller, and was configured as RAID 0. All RAID volumes used hardware RAID.

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

Type of Database

A statement must be provided that describes:

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

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

Microsoft SQL Server 2000 Standard Edition is a relational DBMS.

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

Database Mapping

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

The database was not replicated.

180 Day Space

Details of the 180 day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed.

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

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

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

The details of both the 8-hour transaction log space requirement and the 180-day space requirement is shown in Appendix D.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC	20207.20 tpmC
Price per tpmC	$5.64 per tpmC

Response Times

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

Table 5.2: Response Times

Type	Average	90th %	Maximum
New-Order	0.38	0.54	2.54
Payment	0.21	0.33	2.20
Order-Status	0.26	0.39	2.48
Interactive Delivery	0.10	0.11	0.16
Deferred Delivery	0.77	1.25	4.02
Stock-Level	2.59	3.49	5.53
Menu	0.10	0.11	0.63

Keying and Think Times

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

Table 5.3: Keying Times

Type	Minimum	Average	Maximum
New-Order	18.00	18.01	18.03
Payment	3.00	3.01	3.03
Order-Status	2.00	2.01	2.03
Interactive Delivery	2.00	2.01	2.02
Stock-Level	2.00	2.01	2.02

Table 5.4: Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.10	121.20
Payment	0.00	12.11	121.21
Order-Status	0.00	10.09	100.99
Interactive Delivery	0.00	5.07	50.49
Stock-Level	0.00	5.04	50.49

Response Time Frequency Distribution Curves and Other Graphs

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

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

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for each transaction type.

Keying Time frequency distribution curves (see Clause 5.6.4) must be reported for each transaction type.

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

Figure 3. New Order Response Time Distribution

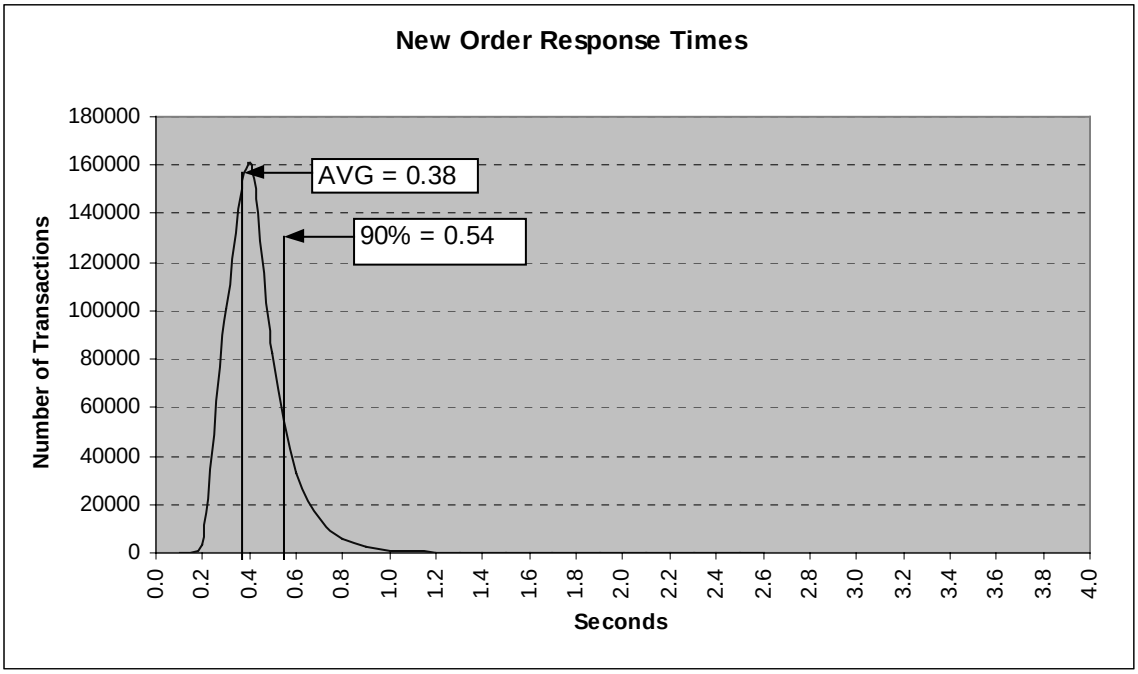


Figure 4. Payment Response Time Distribution

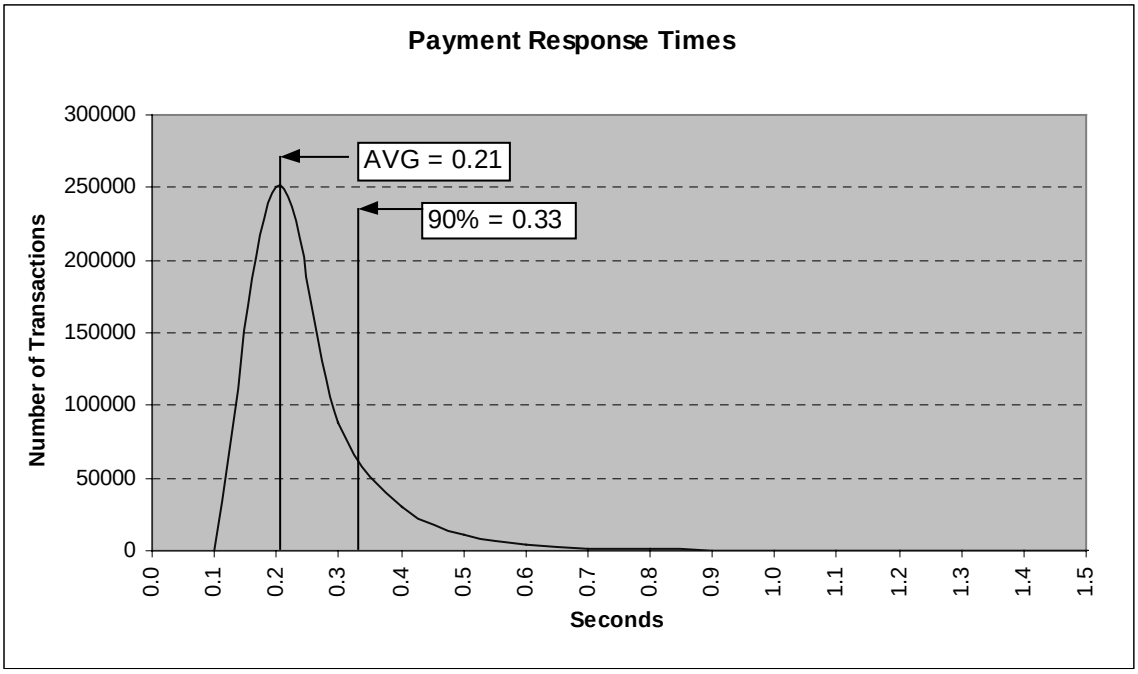


Figure 5. Order Status Response Time Distribution

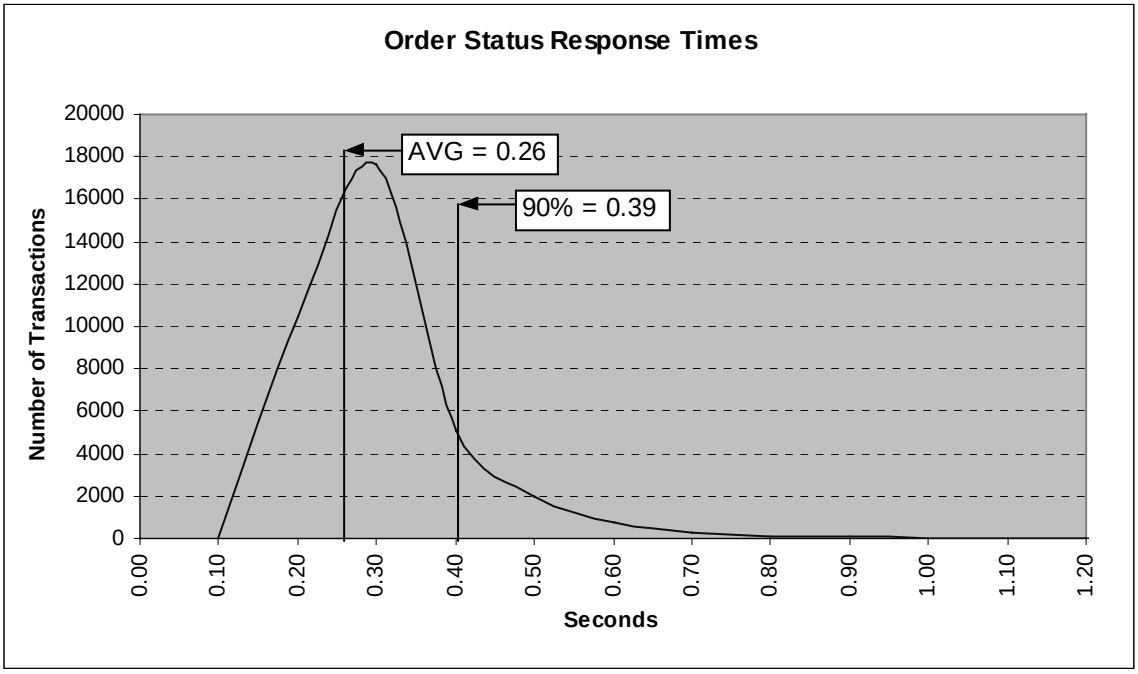


Figure 6. Delivery Response Time Distribution

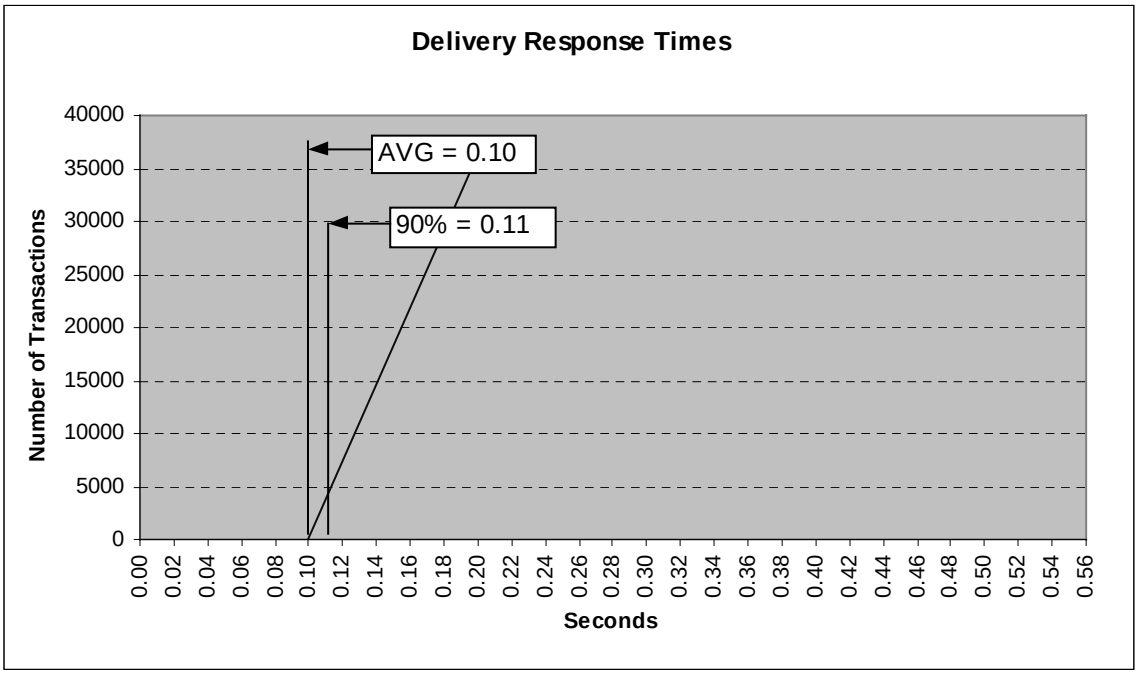


Figure 7. Stock Level Response Time Distribution

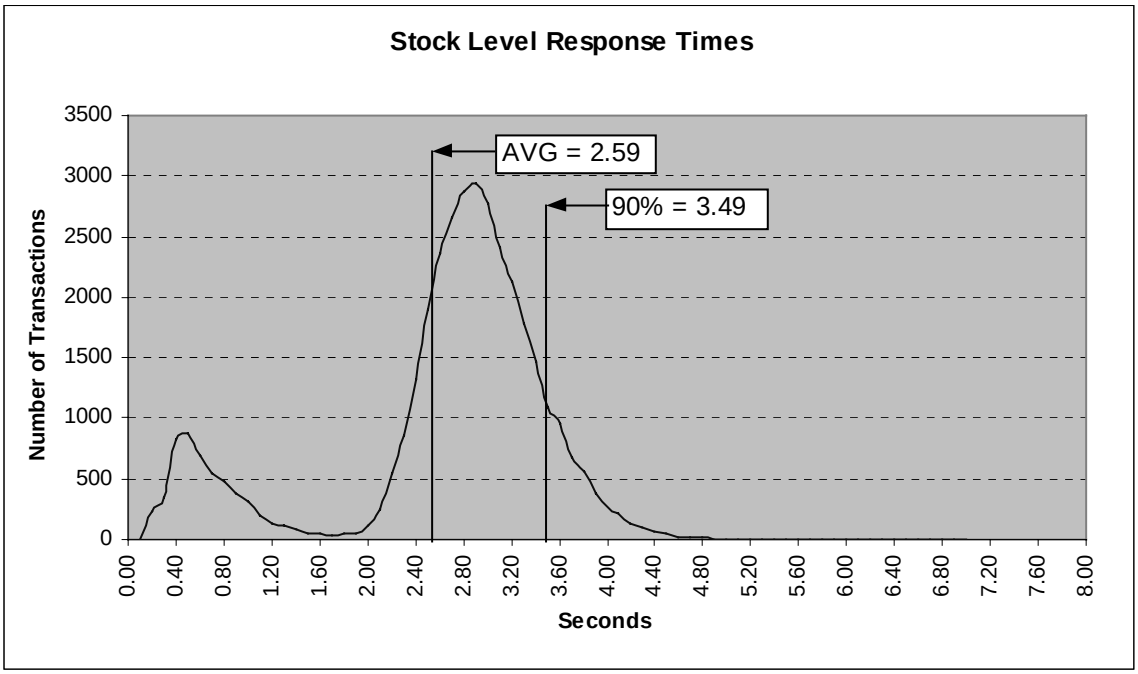


Figure 8. Response Time vs. Throughput

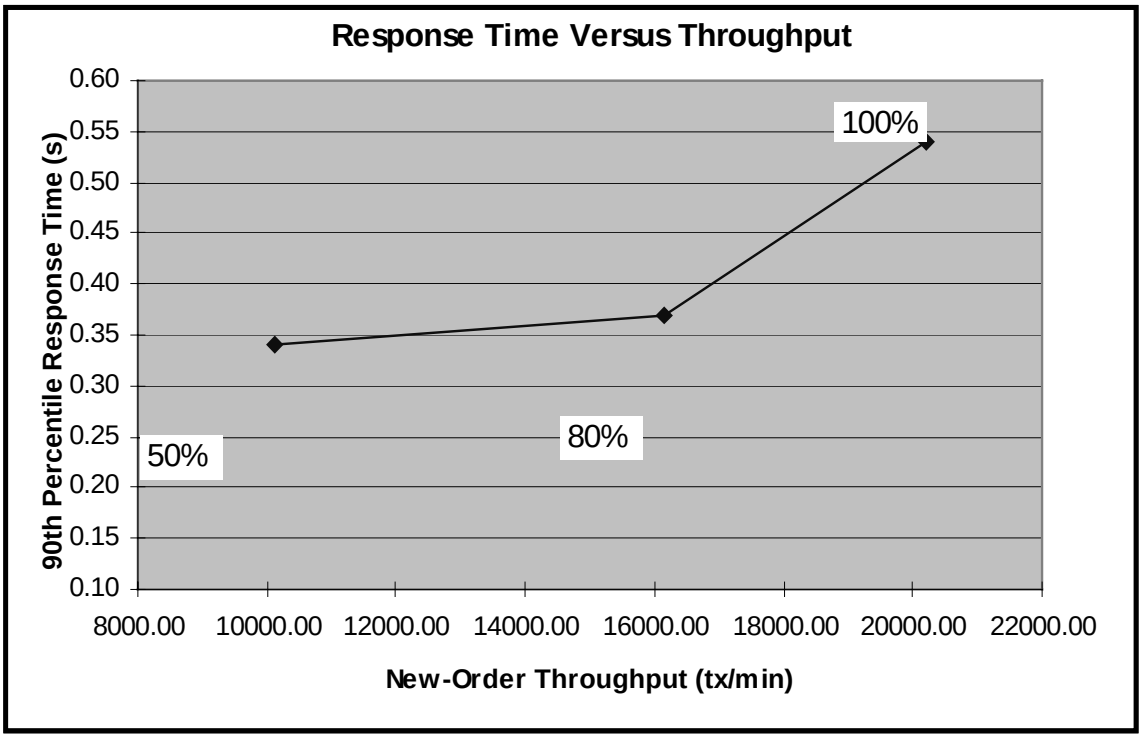


Figure 9. New Order Think Time Distribution

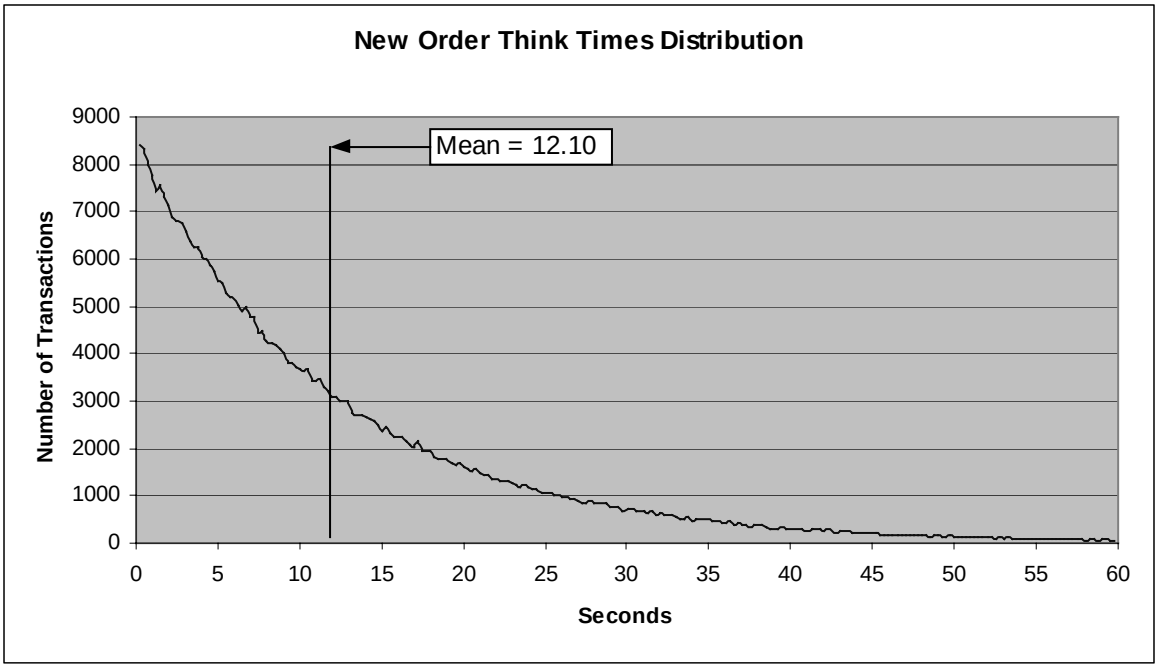
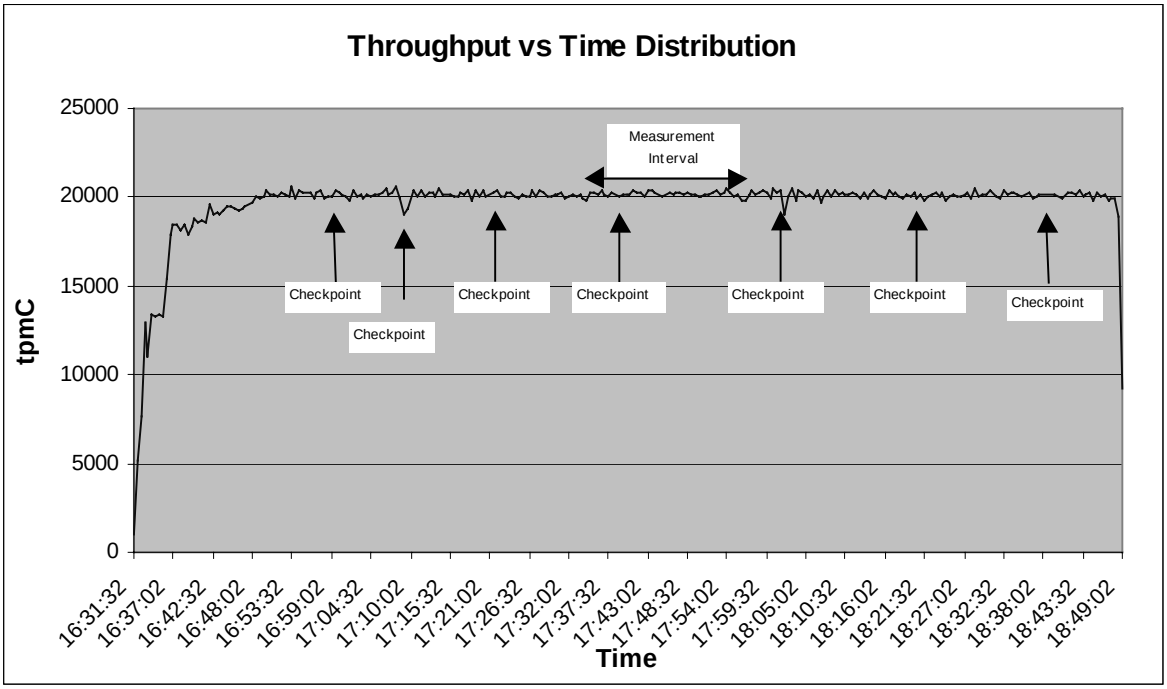


Figure 10. Throughput vs. Time Distribution



Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be disclosed.

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

Work Performed During Steady State

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

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

The RTE generated the required input data for the chosen transaction. It waited to complete the minimum required key time before transmitting the input screen. The transmission was 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.

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

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

To perform checkpoints at specific intervals, we wrote a script to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to the measurement interval which was 20 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. The positioning of the measurement interval was verified to be clear of the guard zones and is depicted on the graph in Figure 10.

Reproducibility

A description of the method used to determine the reproducibility of the measurement results must be reported.

We allowed the database to warm up and to reach a steady state for approximately 90 minutes. The steady state was sustained for a 20-minute (measurement) interval, and was followed by a ramp-down. The repeatable interval result was within 0.398 % of the reported interval result.

Measurement Period Duration

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

The reported measured interval was exactly 20 minutes long.

Regulation of Transaction Mix

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

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

Transaction Statistics

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

Table 5.5: Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.01%
	Remote warehouse order lines	0.99%
	Rolled back transactions	0.98%
	Average items per order	10.00
Payment	Home warehouse payments	85.06%
	Remote warehouse payments	14.94%
	Accessed by last name	60.03%
Delivery	Skipped transactions (interactive)	0
	Skipped transactions (deferred)	0
Order Status	Accessed by last name	59.84%
Transaction Mix	New Order	44.83%
	Payment	43.04%
	Order status	4.04%
	Delivery	4.05%
	Stock level	4.04%

Checkpoint Count and Location

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

The initial checkpoint was started 34 minutes after the start of the ramp-up. Once that checkpoint was complete, SQLServer 2000 automatically began a checkpoint based on the 'recovery interval' database

parameter. Subsequent checkpoints occurred every 20 minutes, beginning at the initial checkpoint. The checkpoint in the measurement interval lasted approximately 9 minutes. The measurement interval contains the forth checkpoint (including the one that SQLServer administrated), and is clear of the guard zones.

Clause 6 Related Items

RTE Descriptions

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

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

Emulated Components

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

The driver system consisted of 2 Compaq ProLiant servers. These driver machines were attached to the 2 Compaq ProLiant client machines through a Gigabit connection to the Gigabit Switches. Since this configuration is the same connectivity of the priced system, no components were being emulated. Therefore, the test described in Clause 6.6.3.4 was not required.

.

Functional Diagrams

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

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

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

Networks

The network configuration of both the tested services and proposed (target) services 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.

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

In the tested configuration, 2 driver (RTE) machines were connected to the 8-port switches with Gigabit Ethernet connections. The 2 client machines were connected to the switch with 100Mb/s connections. The server (SUT) was connected to a switch with a 100Mb/s connection.

Operator Intervention

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

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

Clause 7 Related Items

System Pricing

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

The total 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.

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

Availability, Throughput, and Price Performance

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

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

- **Maximum Qualified Throughput** **20207.20 tpmC**
- **Price per tpmC** **$5.64 per tpmC**
- **Availability** **September 26, 2000**

Country Specific Pricing

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

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

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 3 Microsoft Windows 2000 Server Standard Edition licenses (includes Microsoft COM+)
- 1 Microsoft SQL Server 2000 Standard Edition (3 Processor Licenses)
- 1 Microsoft Visual C++
- Compaq Servers include 3 years of support.

Clause 9 Related Items

Auditor's Report

The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report.

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

Performance Metrics, Inc.
137 Yankton St., Suite 101
Folsom, CA 95630
(phone) (916) 985-1131
(fax) (916) 985-1185
e-mail: lorna@perfmetrics.com

Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
404 Balboa Street
San Francisco, CA 94118 or

Compaq Computer Corporation
Database Performance Engineering
P.O. Box 692000
Houston, TX 77269-2000

PERFORMANCE METRICS INC.
TPC Certified Auditors

July 26, 2001

Mike Nikolaiev
Compaq Computer Corporation
MS 150402
20555 SH 249
Houston, TX 77070

In my opinion the data provided for the ProLiant ML570-700 result of August 21, 2000 complies with the TPC-C Version 5 upgrade requirements.

The following attributes of the benchmark were given special attention:

- The data for the 60 day space calculation was verified
- Maintenance was verified to be 3-year, 7x24 with 4 hour response time
- The system pricing was checked for major components and maintenance

Auditor Notes:

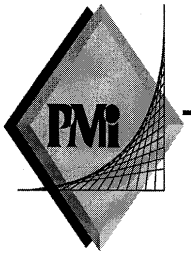
none.

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

137 Yankton Street, Suite 101, Folsom, CA 95630
(916) 985-1131 fax: (916) 985-1185 email: Lorna@PerfMetrics.com



PERFORMANCE METRICS INC.
TPC Certified Auditors

August 18, 2000

Mr. Brean Campbell
Database Performance Engineer
Compaq Computer Corporation
20555 SH 249
Houston, TX 77070

I have verified the TPC Benchmark™ C client/server for the following configuration:

Platform: ProLiant ML570-6/700-3P
Database Manager: Microsoft SQL Server 2000 Standard Edition
Operating System: Microsoft Windows 2000 Standard Edition
Transaction Monitor: Microsoft COM+

Servers: ProLiant ML570 with:				
CPU's	Memory	Disks (total)	90% Response	TpmC
3 Pentium III Xeon @ 700 Mhz	Main: 2560 MB	78 @ 18.2GB 15K rpm 6 @ 18.2GB 10K rpm 1 @ 9.1GB (OS)	0.54 sec	20,207.20
2 Clients: ProLiant ML350's each with:				
1 Pentium III @ 733 Mhz	Main: 384 MB Cache: 256K	1 @ 9.1GB	na	Na

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 1620 warehouses.
- The ACID properties were successfully demonstrated.

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

- Log loss and data loss durability were demonstrated on a subset of the SUT configured with a database properly populated for 10 warehouses.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 180 day space calculation was verified.
- The was no controller cache available on the log disk controllers.
- The steady state portion of the test was 20 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.
- Third party quotes were verified for compliance.

Auditor Notes: None.

Sincerely,

Lorna Livingtree

Lorna Livingtree
Auditor

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

Appendix A: Source Code

The client source code is listed below.

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;}
};
```

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;

        RegCloseKey(hKey);

        return FALSE;
}
```

ReadRegistry.h

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

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

WEBCLNT.DSP

```
# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "webclnt - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

!ELSEIF  "$(CFG)" == "webclnt - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
```

```
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
# End Target
# End Project
```

Webclnt.dsw

```
Microsoft Developer Studio Workspace File, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "install"=.\install\install.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name isapi_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tuxapp
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

###############################################################################

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_encina_dll
    End Project Dependency
}}}

###############################################################################

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp - Package Owner=<4>
```

```
Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
}}}

###############################################################################

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

###############################################################################

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
}}}

###############################################################################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

###############################################################################
```

com_all_resource.h

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

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE         32768
#define _APS_NEXT_CONTROL_VALUE         201
#define _APS_NEXT_SYMED_VALUE           106
#endif
#endif
```

db_dblib_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_dblib_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_dblib_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak" CFG="db_dblib_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_dblib_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "db_dblib_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "db_dblib_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_dblib.dll"

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
```

```
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept
# ADD LINK32 icap.lib ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_dblib_dll - Win32 Release"
# Name "db_dblib_dll - Win32 Debug"
# Name "db_dblib_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_dblib.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

db_odbc_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_odbc_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_odbc_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak" CFG="db_odbc_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_odbc_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "db_odbc_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_odbc.dll"

!ELSEIF  "$(CFG)" == "db_odbc_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
```

```
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept

!ELSEIF  "$(CFG)" == "db_odbc_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_odbc_"
# PROP BASE Intermediate_Dir "db_odbc_"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_odbc_dll - Win32 Release"
# Name "db_odbc_dll - Win32 Debug"
# Name "db_odbc_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_odbc.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

dlldata.c

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

        DO NOT ALTER THIS FILE

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

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

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


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


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )
```

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

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

error.h

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

#pragma once

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

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

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
        int             iError;                         //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_INS_MEMORY                          "Insufficient Memory to continue."
#define ERR_UNKNOWN                             "Unknown error."
#define ERR_MSG_BUF_SIZE              512
#define INV_ERROR_CODE                          -1

class CBaseErr
{
public:
         CBaseErr(LPCTSTR szLoc = NULL)
         {
                  m_idMsg            = INV_ERROR_CODE;

                  if (szLoc)
                  {
                           m_szLoc = new char[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[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;
};

class CSocketErr : public CBaseErr
{
public:
        enum Action
        {
                        eNone,
                        eSend,
                        eSocket,
                        eBind,
                        eConnect,
                        eListen,
                        eHost,
                        eRecv,
        };

        CSocketErr(Action eAction, LPCTSTR szLocation = NULL);
        Action    m_eAction;

        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,
        };

                        CSystemErr(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;}
};
```

install.c

```
/*      FILE:             INSTALL.C
 *                                Microsoft TPC-C Kit Ver. 4.20.000
 *                                Copyright Microsoft, 1999
 *             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
 */

#include <windows.h>
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.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               iThreadTimeout;
static    int               iListenBackLog;
static    int               iAcceptExOutstanding;

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);
static    void              ReadRegistrySettings(void);
static    void              WriteRegistrySettings(char *szDllPath);
static    BOOL              RegisterDLL(char *szFileName);
static    int                       CopyFiles(HWND hDlg, char *szDllPath);
static    BOOL              GetInstallPath(char *szDllPath);
static    void              GetVersionInfo(char *szDLLPath, char *szExePath);
static    BOOL              CheckWWWebService(void);
static    BOOL              StartWWWebService(void);
static    BOOL              StopWWWebService(void);
static    void              UpdateDialog(HWND hDlg);

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             szExePath[256];

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

                        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();

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

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

        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);


        // write binaries to inetpub\wwwroot
        rc = CopyFiles(hDlg, szDllPath);
        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;
        }

        // 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,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
        {
                size = sizeof(iPoolThreadLimit);
                if ( RegQueryValueEx(hKey, "PoolThreadLimit", 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);
        }
}

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 )
        {
                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, 15));
                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)
{
        BOOL            bSvcRunning;

        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);
        }

        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 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_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);

        //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();
        }

        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;
        char      *ptr;
```

```
        int             iRc;

        szDllPath[0] =  0;
        bRc = TRUE;
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
        {
                sv = sizeof(szData);
                iRc = RegQueryValueEx( hKey, "/,", NULL, NULL, szData, &sv );
// used by IIS 3.0
                if (iRc == ERROR_FILE_NOT_FOUND)
                        iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData,
&sv );   // used by IIS 4.0
                if (iRc == ERROR_SUCCESS)
                {
                        bRc = FALSE;
                        strcpy(szDllPath, szData);
                        if ( (ptr = strchr(szDllPath, ',')) )
                                *ptr = 0;

                        len = strlen(szDllPath);
                        if ( szDllPath[len-1] != '\\' )
                        {
                                szDllPath[len] = '\\';
                                szDllPath[len+1] = 0;
                        }
                }

                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);
        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;
}
```

install.h

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

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

#define BN_LOG                                  1001
#define ED_KEEP                                 1002
#define ED_THREADS                              1003
#define ED_THREADS2                             1004
#define IDC_PATH                                1007
#define IDC_VERSION                             1009
#define IDC_RESULTS                             1010
#define IDC_PROGRESS1                           1011
#define IDC_STATUS                              1012
#define IDC_BUTTON1                             1013
#define ED_MAXCONNECTION                        1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT             1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE       1017
#define ED_IIS_THREAD_TIMEOUT                   1018
#define ED_IIS_LISTEN_BACKLOG                   1019
#define IDC_DBLIB                               1021
#define IDC_ODBC                                1022
#define IDC_CONNECT_POOL                                                1023
#define ED_USER_CONNECT_DELAY_TIME                      1024

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

install.rc

```
//Microsoft 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

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

IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION |
    WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT        ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    CONTROL         "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
                    WS_GROUP | WS_TABSTOP,43,100,33,10
    CONTROL         "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,43,113,32,10
    CONTROL         "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,106,100,46,10
    CONTROL         "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
                    WS_DISABLED | WS_TABSTOP,106,113,43,10
    EDITTEXT        ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
    CONTROL         "DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON | WS_GROUP |
                    WS_TABSTOP,45,219,39,12
    CONTROL         "ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON | WS_TABSTOP,
                    91,219,39,12
    EDITTEXT        ED_IIS_MAX_THEAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    DEFPUSHBUTTON   "OK",IDOK,53,331,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,119,331,50,14
    EDITTEXT        IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
    LTEXT           "Max Number of Connections:",IDC_STATIC,35,73,115,12
    RTEXT           "Version 4.11",IDC_VERSION,120,4,89,9
    LTEXT           "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
    LTEXT           "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
                    12
    LTEXT           "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
    LTEXT           "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
    GROUPBOX        "Database Interface",IDC_STATIC,35,208,163,27,WS_GROUP
    LTEXT           "Installation directory:",IDC_STATIC,35,29,71,10
    GROUPBOX        "Transaction Monitor",IDC_STATIC,33,90,165,37
    LTEXT           "Server Name:",IDC_STATIC,35,155,56,8
    LTEXT           "User ID:",IDC_STATIC,35,168,60,8
    LTEXT           "User Password:",IDC_STATIC,35,181,83,8
    LTEXT           "Database Name:",IDC_STATIC,35,194,54,8
    GROUPBOX        "SQL Server Connection Properties",IDC_STATIC,22,139,187,
                    102
    GROUPBOX        "Web Client Properties",IDC_STATIC,22,15,187,118
    GROUPBOX        "IIS Settings",IDC_STATIC,22,247,187,79
    LTEXT           "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

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

IDD_DIALOG3 DIALOG DISCARDABLE  0, 0, 91, 40
STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK | DS_CENTER | WS_CAPTION
CAPTION "Installing TPC-C Web Client"
FONT 12, "Arial Black"
BEGIN
    CONTROL         "Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
                    7,20,77,13
    CTEXT           "Static",IDC_STATUS,7,7,77,12,SS_SUNKEN
END

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


////////////////////////////////////////////////////////////////////////////
//
// DESIGNINFO
//
```

```
#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO DISCARDABLE
BEGIN
    IDD_DIALOG1, DIALOG
    BEGIN
        LEFTMARGIN, 22
        RIGHTMARGIN, 209
        VERTGUIDE, 35
        VERTGUIDE, 198
        TOPMARGIN, 4
        BOTTOMMARGIN, 345
    END

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

    IDD_DIALOG3, DIALOG
    BEGIN
        LEFTMARGIN, 7
        RIGHTMARGIN, 84
        TOPMARGIN, 7
        BOTTOMMARGIN, 33
    END

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


#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


////////////////////////////////////////////////////////////////////////////
//
// Icon
//

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

////////////////////////////////////////////////////////////////////////////
//
// TPCCDLL
//

IDR_TPCCDLL             TPCCDLL DISCARDABLE     "..\\..\\isapi_dll\\bin\\tpcc.dll"

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

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

#endif    // !_MAC
```

```
/////////////////////////////////////////////////////////////////////////////
//
// LICENSE
//

IDR_LICENSE1            LICENSE DISCARDABLE     "license.txt"

/////////////////////////////////////////////////////////////////////////////
//
// DBLIB_DLL
//

IDR_DBLIB_DLL           DBLIB_DLL DISCARDABLE
"..\\..\\db_dblib_dll\\bin\\tpcc_dblib.dll"

/////////////////////////////////////////////////////////////////////////////
//
// ODBC_DLL
//

IDR_ODBC_DLL            ODBC_DLL DISCARDABLE
"..\\..\\db_odbc_dll\\bin\\tpcc_odbc.dll"

/////////////////////////////////////////////////////////////////////////////
//
// TUXEDO_APP
//

IDR_TUXEDO_APP          TUXEDO_APP DISCARDABLE  "..\\..\\tuxapp\\bin\\tuxapp.exe"

/////////////////////////////////////////////////////////////////////////////
//
// TUXEDO_DLL
//

IDR_TUXEDO_DLL          TUXEDO_DLL DISCARDABLE
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"

/////////////////////////////////////////////////////////////////////////////
//
// COM_DLL
//

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

/////////////////////////////////////////////////////////////////////////////
//
// COM_PS_DLL
//

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

/////////////////////////////////////////////////////////////////////////////
//
// COM_ALL_DLL
//

IDR_COMALL_DLL          COM_ALL_DLL DISCARDABLE
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"
#endif    // English (U.S.) resources
```

```
/////////////////////////////////////////////////////////////////////////////


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


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

install_com.cpp

```
/*        FILE:              INSTALL_COM.CPP
 *                                   Microsoft TPC-C Kit Ver. 4.20.000
 *                                   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 =        "";
        // 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;
}
```

install_resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1                     101
#define IDI_ICON1                       102
#define IDR_TPCCDLL                     103
#define IDD_DIALOG2                     105
#define IDI_ICON2                       106
#define IDR_DELIVERY                    107
#define IDD_DIALOG3                     108
#define IDR_LICENSE1                    112
#define IDD_DIALOG4                     113
#define IDR_TPCCOBJ1                    117
#define IDR_TPCCSTUB1                   118
#define IDR_DBLIB_DLL                   122
#define IDR_ODBC_DLL                    123
#define IDR_TUXEDO_APP                  124
#define IDR_TUXEDO_DLL                  125
#define IDR_COM_DLL                     126
#define IDR_COMPS_DLL                   127
#define IDR_COMALL_DLL                  128
#define BN_LOG                          1001
#define ED_KEEP                         1002
#define ED_THREADS                      1003
#define ED_THREADS2                     1004
#define IDC_PATH                        1007
#define IDC_VERSION                     1009
#define IDC_RESULTS                     1010
#define IDC_PROGRESS1                   1011
#define IDC_STATUS                      1012
#define IDC_BUTTON1                     1013
#define ED_MAXCONNECTION                1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT     1015
#define ED_MAXDELIVERIES                1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT           1018
#define ED_IIS_LISTEN_BACKLOG           1019
#define IDC_DBLIB                       1021
#define IDC_LICENSE                     1022
#define IDC_ODBC                        1022
#define IDC_CONNECT_POOL                1023
#define ED_DB_SERVER                    1023
#define ED_USER_CONNECT_DELAY_TIME      1024
#define ED_DB_USER_ID                   1024
#define IDC_MTS                         1025
#define IDC_TM_MTS                      1025
#define IDC_TM_TUXEDO                   1026
#define IDC_TM_NONE                     1027
#define ED_DB_PASSWORD                  1028
#define ED_DB_NAME                      1029
#define IDC_TM_ENCINA                   1030

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        129
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1024
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

isapi_dll.dsp

```
# Microsoft Developer Studio Project File - Name="isapi_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=isapi_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak" CFG="isapi_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
```

```
!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "isapi_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/nodefaultlib:"LIBCMT" /out:".\bin/tpcc.dll"
# SUBTRACT LINK32 /nodefaultlib

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib
..\common\txnlog\lib\debug\spinlock.lib ..\common\txnlog\lib\debug\error.lib
..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/nodefaultlib:"LIBCMTD" /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /nodefaultlib

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dl"
# PROP BASE Intermediate_Dir "isapi_dl"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll"
/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target
```

```
# Name "isapi_dll - Win32 Release"
# Name "isapi_dll - Win32 Debug"
# Name "isapi_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.def, *.rc"
# Begin Source File

SOURCE=.\src\tpcc.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.rc
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "*.h, *.hpp"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\common\src\ReadRegistry.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.h
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\src\tpcc_tux.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

isapi_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
```

rtetime.h

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

//FILE: RTETIME.H

#define   MAX_JULIAN_TIME              0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME__int64
#define   TC_TIME            DWORD
extern "C"
{
BOOL               InitJulianTime(LPSYSTEMTIME lpInitTime);
JULIAN_TIME        GetJulianTime(void);
DWORD              MyTickCount(void);
void               GetJulianAndTC(JULIAN_TIME *pJulian, DWORD *pTC);
JULIAN_TIME        ConvertTo64BitTime(int iYear, int iMonth, int iDay, int iHour,
int iMinute, int iSecond);
JULIAN_TIME        Get64BitTime(LPSYSTEMTIME lpInitTime);
int                         JulianDay( int yr, int mm, int dd );
void               JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd,
int *hh, int *mi, int *ss );
void               JulianToCalendar( int day, int* yr, int* mm, int* dd );
}
```

spinlock.h

```
/*       FILE: SPINLOCK.H
 *
 * Copyright 1997 Microsoft Corp., All rights reserved.
 *
 * Source code licensed to Tandem Computers  for Internal
 * use only.  Redistribution of source or object files or
 * any derivative works is prohibited. By agreement, this
 * notice may not be removed.
```

```
 *
 * Authors: Mike Parkes, Charles Levine, Philip Durr
 *                          Microsoft Corp.
 */

#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
```

tm_com_dll.dsp

```
# Microsoft Developer Studio Project File - Name="tm_com_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tm_com_dll - Win32 Debug
```

```
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak" CFG="tm_com_dll - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tm_com_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tm_com_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tm_com_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_com.dll"

!ELSEIF  "$(CFG)" == "tm_com_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_com.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "tm_com_dll - Win32 Release"
# Name "tm_com_dll - Win32 Debug"
# Begin Source File

SOURCE=.\src\tpcc_com.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com.h
# End Source File
# End Target
# End Project
```

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];

         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

         int                                 iRetryCnt = 0;
         static int                  iMaxRetries = 10;

         assert(txnDelilog != NULL);

Reconnect:
         try
         {
                  if (Reg.eDB_Protocol == ODBC)
                           pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                  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;

                  // will retry connection up to ten times
                  if (iRetryCnt++ < iMaxRetries)
                  {
                           Sleep(5000);       // delay for 5 seconds
                           goto Reconnect;
                  }

                  wsprintf( szTmp, "Delivery Txn thread terminating after %d
retries.", iMaxRetries );
                  WriteMessageToEventLog( szTmp );
                  goto ErrorExit;
         }
         catch (...)
         {
                  WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread.  Delivery Txn thread terminating."));
                  goto ErrorExit;
         }


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

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

```
                                        // make a local copy of current entry from
delivery buffer and increment buffer index

        EnterCriticalSection(&DelBuffCriticalSection);
                                        delivery = *(pDelBuff+dwDelBuffBusyIndex);
                                        dwDelBuffFreeCount++;
                                        dwDelBuffBusyIndex++;
                                        if (dwDelBuffBusyIndex == dwDelBuffSize)
        // wrap-around if at end of buffer
                                                dwDelBuffBusyIndex = 0;


        LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

                                        //log txn
                                        txnDeliRec.TxnStatus = ERR_SUCCESS;
                                        for (int i=0; i<10; i++)
                                                txnDeliRec.o_id[i] =
pDeliveryData->o_id[i];
                                        txnDeliRec.DeltaT4 =
(int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartT0);
                                        txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                        if (txnDelilog != NULL)
                                                txnDelilog-
>WriteToLog(&txnDeliRec);
                        }
                }
                catch (CBaseErr *e)
                {
                        char szTmp[1024];
                        wsprintf( szTmp, "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(short 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=4><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 );
                else if (Reg.eDB_Protocol == DBLIB)
                        Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
        }
        catch (...)
        {
                TermDelete(iNewTerm);
                throw;              // pass exception upward
        }

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

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

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

         EnterCriticalSection(&TermCriticalSection);

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

         LeaveCriticalSection(&TermCriticalSection);

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

char *CWEBCLNT_ERR::ErrorText()
{
         static SERRORMSG errorMsgs[] =
         {
                  {        ERR_COMMAND_UNDEFINED,
                  "Command undefined."                                                    },

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

```
        {       ERR_ORDERSTATUS_CLT_RANGE,
"Order Status Customer last name longer than 16 characters."
},
        {       ERR_ORDERSTATUS_DID_INVALID,
"Order Status District invalid, value must be numeric 1 - 10."
},
        {       ERR_ORDERSTATUS_MISSING_CID_CLT,
"Order Status Either Customer ID or Last Name must be entered."
},
        {       ERR_ORDERSTATUS_MISSING_CID_KEY,
"Order Status missing Customer key \"CID*\"."
                },
        {       ERR_ORDERSTATUS_MISSING_CLT_KEY,
"Order Status missing Customer Last Name key \"CLT*\"."
        },
        {       ERR_ORDERSTATUS_MISSING_DID_KEY,
"Order Status missing District key \"DID*\"."
                },
        {       ERR_PAYMENT_CDI_INVALID,
"Payment Customer district invalid must be numeric."
        },
        {       ERR_PAYMENT_CID_AND_CLT,
"Payment Only Customer ID or Last Name may be entered, not both."  },
        {       ERR_PAYMENT_CUSTOMER_INVALID,
"Payment Customer data type invalid, must be numeric."
        },
        {       ERR_PAYMENT_CWI_INVALID,
"Payment Customer Warehouse invalid, must be numeric."
        },
        {       ERR_PAYMENT_DISTRICT_INVALID,
"Payment District ID is invalid, must be 1 - 10."
        },
        {       ERR_PAYMENT_HAM_INVALID,
"Payment Amount invalid data type must be numeric."
                },
        {       ERR_PAYMENT_HAM_RANGE,
        "Payment Amount out of range, 0 - 9999.99."
                        },
        {       ERR_PAYMENT_LAST_NAME_TO_LONG,
"Payment Customer last name longer than 16 characters."
        },
        {       ERR_PAYMENT_MISSING_CDI_KEY,
"Payment missing Customer district key \"CDI*\"."
        },
        {       ERR_PAYMENT_MISSING_CID_CLT,
"Payment Either Customer ID or Last Name must be entered."
},
        {       ERR_PAYMENT_MISSING_CID_KEY,
"Payment missing Customer Key \"CID*\"."
                        },
        {       ERR_PAYMENT_MISSING_CLT_KEY,
"Payment missing Customer Last Name key \"CLT*\"."
                },
        {       ERR_PAYMENT_MISSING_CWI_KEY,
"Payment missing Customer Warehouse key \"CWI*\"."
                },
        {       ERR_PAYMENT_MISSING_DID_KEY,
"Payment missing District Key \"DID*\"."
                        },
        {       ERR_PAYMENT_MISSING_HAM_KEY,
"Payment missing Amount key \"HAM*\"."
                        },
        {       ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,  "Stock Level;
missing Threshold key \"TT*\"."
        },
        {       ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in the range = 1 - 99."
        },
        {       ERR_STOCKLEVEL_THRESHOLD_RANGE,
        "Stock Level Threshold out of range, range must be 1 - 99."
        },
        {       ERR_VERSION_MISMATCH,
                "Invalid version field.  RTE and Web Client are probably out of
sync." },
                {       ERR_W_ID_INVALID,
                "Invalid Warehouse ID."
                                                                        },
                {       0,
                                        ""

                                                        }
        };

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

        if (m_szTextDetail)
                strcat( szTmp, m_szTextDetail );
        if (m_SystemErr)
                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: %4.4d   District: %2.2d<BR> <BR>",
                  STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
                  Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

         if ( bInput )
         {
                  strcpy(szForm+c,
                           "Stock Level Threshold: <INPUT NAME=\"TT*\"
SIZE=2><BR> <BR>"
                           "low stock:    </font><BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR>"
                           " <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"
                           "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                           "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                           "</FORM></HTML>" );
         }
         else
         {
                  wsprintf(szForm+c,
                           "Stock Level Threshold: %2.2d<BR> <BR>"
                           "low stock: %3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR>"
                           " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></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: %4.4d   ",
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: %4.4d   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, "  %4.4d   %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: %4.4d"
                        "                                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: %4.4d
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: %4.4d  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> <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: %4.4d   ",
                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></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, "  %4.4d       %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: %4.4d<BR> <BR>",
                (!bInput && (pDeliveryData->exec_status_code != eOK)) ?
ERR_TYPE_DELIVERY_POST : 0,
                DELIVERY_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

        if ( bInput )
        {
                strcpy( szForm+c,
```

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

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

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

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

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

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

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

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

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

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

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

tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

        GetExtensionVersion@1
```

```
        HttpExtensionProc  @2
        TerminateExtension @3
```

tpcc.h

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

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

#define TP_MAX_RETRIES
        50


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

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


//This structure defines the data necessary to keep distinct for each terminal or
client connection.
typedef struct _CLIENTDATA
{
        int                         iNextFree;
        //index of next free element or -1 if this entry in use.
        int                         w_id;
        //warehouse id assigned at welcome form
        int                         d_id;
        //district id assigned at welcome form
        int                         iSyncId;
        //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(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
```

```
void DeliveryWorkerThread(void *ptr);
```

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
```

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 );
}
```

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.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.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.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_all" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tpcc_com_all - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_all - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "tpcc_com_all - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
```

```
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /machine:I386

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "tpcc_com_all - Win32 Release"
# Name "tpcc_com_all - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.c"
# Begin Source File

SOURCE=.\src\tpcc_com_all.cpp
# SUBTRACT CPP /YX
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.idl

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
          midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
          midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\src\Methods.h
# End Source File
# Begin Source File

SOURCE=.\src\resource.h
# End Source File
# End Group
# Begin Source File

SOURCE=.\src\tpcc_com_all.rc
# End Source File
# End Target
# End Project
```

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 Mon Jun 12 18:15:19 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.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.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.rgs

```
HKCR
{
         TPCC.AllTxns.1 = s 'All Txns Class'
         {
                  CLSID = s '{122A3128-2520-11D3-BA71-00C04FBFE08B}'
         }
         TPCC.AllTxns = s 'TPCC Class'
         {
                  CurVer = s 'TPCC.AllTxns.1'
         }
         NoRemove CLSID
         {
                  ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC
Class'
                  {
                           ProgID = s 'TPCC.AllTxns.1'
                           VersionIndependentProgID = s 'TPCC.AllTxns'
                           InprocServer32 = s '%MODULE%'
                           {
                                    val ThreadingModel = s 'Both'
                           }
                  }
         }
}
```

tpcc_com_all_i.c

```
#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 Mon Jun 12 18:15:19 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 Mon Jun 12 18:15:19 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_no.rgs

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

tpcc_com_os.rgs

```
HKCR
{
	TPCC.OrderStatus.1 = s 'OrderStatus Class'
	{
		CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
	}
	TPCC.OrderStatus = s 'OrderStatus Class'
	{
		CurVer = s 'TPCC.OrderStatus.1'
	}
	NoRemove CLSID
	{
		ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s
'OrderStatus Class'
		{
			ProgID = s 'TPCC.OrderStatus.1'
			VersionIndependentProgID = s 'TPCC.OrderStatus'
			InprocServer32 = s '%MODULE%'
			{
				val ThreadingModel = s 'Both'
			}
		}
	}
}
```

tpcc_com_pay.rgs

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

tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
	DllGetClassObject       @1   PRIVATE
	DllCanUnloadNow         @2   PRIVATE
	GetProxyDllInfo         @3   PRIVATE
	DllRegisterServer            @4        PRIVATE
	DllUnregisterServer          @5        PRIVATE
```

tpcc_com_ps.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_ps" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
```

```
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
         copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86 /def:".\src\tpcc_com_ps.def"
/pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
         copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

SOURCE=.\src\dlldata.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.idl

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl
```

```
BuildCmds= \
         midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
         midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project
```

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 Mon Jun 12 18:15:12 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.idl

```
/*       FILE:              ITPCC.IDL
 *                                    Microsoft TPC-C Kit Ver. 4.20.000
 *                                    Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                    not yet audited
 *
 *       PURPOSE:  Defines the interface used by TPCC. This interface can be
implemented by C++ components.
 *
 *  Change history:
 *                 4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
```

```
import "oaidl.idl";
import "ocidl.idl";

         [
                  object,
                  oleautomation,
                  uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                  helpstring("ITPCC Interface"),
                  pointer_default(unique)
         ]
         interface ITPCC : IUnknown
         {

        HRESULT _stdcall NewOrder
                                                  (
                                                  [in] VARIANT  txn_in,
                                                  [out] VARIANT *txn_out
                                                  );

        HRESULT _stdcall Payment
                                                  (
                                                  [in] VARIANT txn_in,
                                                  [out] VARIANT *txn_out
                                                  );

        HRESULT _stdcall Delivery
                                                  (
                                                  [in] VARIANT txn_in,
                                                  [out] VARIANT *txn_out
                                                  );

        HRESULT _stdcall StockLevel
                                                  (
                                                  [in] VARIANT txn_in,
                                                  [out] VARIANT *txn_out
                                                  );

        HRESULT _stdcall OrderStatus
                                                  (
                                                  [in] VARIANT txn_in,
                                                  [out] VARIANT *txn_out
                                                  );

        HRESULT _stdcall CallSetComplete
                                                  (
                                                  );

    };   // interface ITPCC
```

tpcc_com_ps_i.c

```
#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 Mon Jun 12 18:15:12 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 Mon Jun 12 18:15:12 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_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 Mon Jun 12 18:15:12 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,0x47,0x00,0xC0,0x4F,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,
    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 5.3.280 */
    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( 0xffffffe0 ),    /* 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( 0xffffffe0 ),    /* 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( 0xfffffd4 ),    /* 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( 0xfffffd4 ),    /* 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( 0xfffffd4 ),    /* 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( 0xfffffff0c ),    /* 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( 0xffffffe8 ),    /* 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( 0xfffffff8 ),    /* 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( 0xffffffee ),    /* Offset= -18 (836) */
/* 856 */ 0x5c,              /* FC_PAD */
                            0x5b,               /* FC_END */
/* 858 */
                            0x1a,               /* FC_BOGUS_STRUCT */
                            0x3,                /* 3 */
/* 860 */ NdrFcShort( 0x28 ),/* 40 */
/* 862 */ NdrFcShort( 0xffffffee ),    /* 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( 0xffffffef6 ),    /* 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( 0xffffffc42 ),    /* 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( 0xffffffc32 ),    /* 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 Mon Jun 12 18:15:12 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 5.3.280 */
    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( 0xfffffffc ),    /* 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( 0xfffffffc ),    /* 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( 0xfffffffc ),    /* 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( 0xfffffe0 ),    /* 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( 0xfffffff7 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffff02 ),    /* 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( 0xffffffdc ),    /* 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)*/
```

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_dblib.cpp

```
/*      FILE:               TPCC_DBLIB.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:  Implements dblib calls for TPC-C txns.
 *       Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  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);

        // set connection properties to match those used by ODBC
        dbcmd(m_dbproc, "set ANSI_DEFAULTS ON ");
        dbcmd(m_dbproc, "set CURSOR_CLOSE_ON_COMMIT OFF ");
        dbcmd(m_dbproc, "set IMPLICIT_TRANSACTIONS OFF ");
        dbcmd(m_dbproc, "set NOCOUNT ON ");                     // do not
return row counts
        dbcmd(m_dbproc, "set XACT_ABORT ON ");          // rollback transaction
on abort

        // for coyote
//      dbcmd(m_dbproc, "set ansi_warnings on ");       //
//      dbcmd(m_dbproc, "set ansi_nulls on ");          //

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

        // This value must match the number of commands above.
//      DiscardNextResults(2);
        DiscardNextResults(5);      // coyote

        // 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, SQLINT2, -1, -1, (BYTE
*) &m_txn.StockLevel.w_id);     // @w_id smallint
                        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, SQLINT2, -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, SQLINT2, -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, SQLINT2, -1, -1, (BYTE
*) &m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -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, SQLINT2, -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, SQLINT2, -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;
}
```

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."
			ERR_RETRIED_TRANS,			// "Retries
before transaction succeeded."
		};

		CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0;
};

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

		int		m_errno;
		int		m_iTryCount;

		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);
```

tpcc_odbc.cpp

```
/*        FILE:             TPCC_ODBC.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:  Implements ODBC calls for TPC-C txns.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                4.20.000 - updated rev number to match kit
 *                4.10.001 - not deleting error class in catch handler on deadlock
retry;
 *                                      not a functional bug, but a memory leak
 */

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

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>
#include <odbcss.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_odbc.h"

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

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

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

static SQLHENV henv = SQL_NULL_HENV;                            // ODBC
environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
        switch( ul_reason_for_call )
        {
                case DLL_PROCESS_ATTACH:
                        DisableThreadLibraryCalls(hModule);
                        if ( SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                                return FALSE;
```

```
                        break;

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

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

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

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

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

                                                }
        };

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
        LPCSTR szServer,            // name of SQL server
        LPCSTR szUser,                      // user name for login
        LPCSTR szPassword,          // password for login
        LPCSTR szHost,                      // not used
        LPCSTR szDatabase )         // name of database to use
{
        return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase );
}


CTPCC_ODBC::CTPCC_ODBC (
        LPCSTR szServer,                            // name of SQL server
        LPCSTR szUser,                                          // user name
for login
        LPCSTR szPassword,                          // password for login
        LPCSTR szHost,                                  // not used
        LPCSTR szDatabase                           // name of database to
use
        )
{
        RETCODE            rc;

        // initialization
        m_hdbc = SQL_NULL_HDBC;
        m_hstmt = SQL_NULL_HSTMT;

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

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


        if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) != SQL_SUCCESS )
                ThrowError(CODBCERR::eAllocHandle);

        if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )
                ThrowError(CODBCERR::eConnOption);

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

                sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                        szServer, szUser, szPassword, szDatabase );

                rc = SQLDriverConnect(m_hdbc, NULL, (SQLCHAR*)szConnectStr,
sizeof(szConnectStr),
                        (SQLCHAR*)szOutStr, sizeof(szOutStr), &iOutStrLen,
SQL_DRIVER_NOPROMPT );

                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eConnect);
        }

        if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) != SQL_SUCCESS)
                ThrowError(CODBCERR::eAllocHandle);

        {
                char               buffer[128];

                // set some options affecting connection behavior
                strcpy(buffer, "set nocount on ");
                strcat(buffer, "set XACT_ABORT ON " );

                // for coyote
                strcat(buffer, "set ansi_warnings on " );
                strcat(buffer, "set ansi_nulls on " );
```

```
                rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eExecDirect);

                // verify that version of stored procs on server is correct
                char db_sp_version[10];
                strcpy(buffer, "{call tpcc_version}");
                rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eExecDirect);
                if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR, &db_sp_version,
sizeof(db_sp_version), NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
                if ( SQLFetch(m_hstmt) == SQL_ERROR )
                        ThrowError(CODBCERR::eFetch);
                if (strcmp(db_sp_version,sVersion))
                        throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );

                SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt);
        }

        // Bind parameters for each of the transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
        // note: descriptors are automatically released when the connection is
dropped
        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus);
        SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel);

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

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
        RETCODE            rc;
        SDWORD             lNativeError;
        char               szState[6];
        char               szMsg[SQL_MAX_MESSAGE_LENGTH];
        char               szTmp[6*SQL_MAX_MESSAGE_LENGTH];
        CODBCERR   *pODBCErr;                           // not allocated until
needed (maybe never)

        pODBCErr = new CODBCERR();

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

        szTmp[0] = 0;
        while (TRUE)
        {
                rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE *)&szState,
&lNativeError,
                                                (BYTE *)&szMsg, sizeof(szMsg),
NULL);
                if (rc == SQL_NO_DATA)
                        break;

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

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

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

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

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

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

        SQLFreeStmt(m_hstmt, SQL_CLOSE);
        throw pODBCErr;
}

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

        m_hstmt = m_hstmtStockLevel;

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

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

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

        m_hstmt = m_hstmtStockLevel;

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_stocklevel(?,?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::eExecDirect);

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

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

        m_hstmt = m_hstmtNewOrder;

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

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

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

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

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name, sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.NewOrder.OL[0].ol_stock, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_i_price, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);
#else
        // prototype to eliminate patindex in server; shift work to client
        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_ol_i_name,
sizeof(m_ol_i_name), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT, &m_ol_stock, 0, NULL)
!= SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_i_data,
sizeof(m_i_data), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_s_data,
sizeof(m_s_data), NULL) != SQL_SUCCESS
```

```
                    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_i_price, 0,
NULL) != SQL_SUCCESS
                    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_amount, 0, NULL)
!= SQL_SUCCESS
                    )
                    ThrowError(CODBCERR::eBindCol);
#endif

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

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

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

                     // 0         1         2

                     // 01234567890123456789012345678 9
          wchar_t                                   szSqlTemplate[] = L"{call
tpcc_neworder(?,?,?,?,?,?,"

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

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

                       L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}";

          m_hstmt = m_hstmtNewOrder;

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

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

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

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

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

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

                                        strcpy( m_txn.NewOrder.OL[i].ol_i_name,
m_ol_i_name );

                                        if ( strstr(m_i_data, "ORIGINAL") != NULL &&
strstr(m_s_data, "ORIGINAL") != NULL )

          m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'B';
                                        else

          m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'G';
                                        m_txn.NewOrder.OL[i].ol_brand_generic[1] =
0;

                                        m_txn.NewOrder.OL[i].ol_stock
          = m_ol_stock;
                                        m_txn.NewOrder.OL[i].ol_i_price
                    = m_ol_i_price;
                                        m_txn.NewOrder.OL[i].ol_amount
                    = m_ol_amount;
#endif

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

          ThrowError(CODBCERR::eMoreResults);
```

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

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

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

                                SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

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

                m_hstmt = m_hstmtPayment;

                int i = 0;
                if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_DOUBLE,
SQL_NUMERIC, 6, 2, &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
                        || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindParam);

                i = 0;
                if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,     &m_txn.Payment.c_id,
                0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_last,                sizeof(m_txn.Payment.c_last), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.h_date,                0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_1,   sizeof(m_txn.Payment.w_street_1), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2,   sizeof(m_txn.Payment.w_street_2), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city,                sizeof(m_txn.Payment.w_city), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state,               sizeof(m_txn.Payment.w_state), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,                 sizeof(m_txn.Payment.w_zip), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1,   sizeof(m_txn.Payment.d_street_1), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2,   sizeof(m_txn.Payment.d_street_2), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,                sizeof(m_txn.Payment.d_city), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,               sizeof(m_txn.Payment.d_state), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,                 sizeof(m_txn.Payment.d_zip), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,               sizeof(m_txn.Payment.c_first), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,     sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1,   sizeof(m_txn.Payment.c_street_1), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2,   sizeof(m_txn.Payment.c_street_2), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,                sizeof(m_txn.Payment.c_city), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,               sizeof(m_txn.Payment.c_state), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,                 sizeof(m_txn.Payment.c_zip), NULL) !=
SQL_SUCCESS
```

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

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

        m_hstmt = m_hstmtPayment;

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

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_payment(?,?,?,?,?,?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::eExecDirect);

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

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

                        if (m_txn.Payment.c_id == 0)
                                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                        else
                                m_txn.Payment.exec_status_code = eOK;

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

                        // hit deadlock; backoff for increasingly longer
period
                        delete e;
                        Sleep(10 * iTryCount);
                }
        }
//        if (iTryCount)
//                throw new CTPCC_ODBC_ERR(CTPCC_ODBC_ERR::ERR_RETRIED_TRANS,
iTryCount);
}

void CTPCC_ODBC::InitOrderStatusParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtOrderStatus) !=
SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtOrderStatus;

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

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

        // configure block cursor
        if (   SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
                || SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);


        if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);
```

```
        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0,
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.o_entry_d, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.o_carrier_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.c_balance, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);
}

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

        m_hstmt = m_hstmtOrderStatus;

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

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

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

                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_orderstatus(?,?,?,?)}", SQL_NTS);
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                                ThrowError(CODBCERR::eExecDirect);

                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )
                                ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLFetchScroll( m_hstmt, SQL_FETCH_NEXT, 0 );
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                                ThrowError(CODBCERR::eFetchScroll);

                        m_txn.OrderStatus.o_ol_cnt = (short)m_RowsFetched;
                        if (m_txn.OrderStatus.o_ol_cnt != 0)
                        {
                                if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::eSetStmtAttr);

                                if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
        ThrowError(CODBCERR::eMoreResults);

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

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

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

        m_hstmt = m_hstmtDelivery;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

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

```
        {
                if ( SQLBindCol(m_hstmt, (UWORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
        }
}

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

        m_hstmt = m_hstmtDelivery;

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_delivery(?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::eExecDirect);

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

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

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

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

tpcc_odbc.h

```
/*      FILE:               TPCC_ODBC.H
 *                                  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 CODBCERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eAllocConn,                     // error from
SQLAllocConnect
                        eAllocHandle,           // error from
SQLAllocHandle
                        eConnOption,            // error from
SQLSetConnectOption
                        eConnect,               // error from SQLConnect
                        eAllocStmt,                     // error from
SQLAllocStmt
                        eExecDirect,            // error from
SQLExecDirect
                        eBindParam,                     // error from
SQLBindParameter
                        eBindCol,               // error from SQLBindCol
                        eFetch,                         // error from
SQLFetch
                        eFetchScroll,           // error from
SQLFetchScroll
                        eMoreResults,           // error from
SQLMoreResults
                        ePrepare,               // error from SQLPrepare
                        eExecute,               // error from SQLExecute
                        eSetEnvAttr,            // error from
SQLSetEnvAttr
                        eSetStmtAttr            // error from
SQLSetStmtAttr
                };

                CODBCERR(void)
                {
                        m_eAction = eNone;
                        m_NativeError = 0;
                        m_bDeadLock = FALSE;
                        m_odbcerrstr = NULL;
                };

                ~CODBCERR()
                {
                        if (m_odbcerrstr != NULL)
                                delete [] m_odbcerrstr;
                };

                ACTION    m_eAction;
                int               m_NativeError;
                BOOL      m_bDeadLock;
                char   *m_odbcerrstr;

                int ErrorType() {return ERR_TYPE_ODBC;};
```

```
                int ErrorNum() {return m_NativeError;};
                char *ErrorText() {return m_odbcerrstr;};

};

class CTPCC_ODBC_ERR : public CBaseErr
{
        public:
                enum TPCC_ODBC_ERRS
                {
                        ERR_WRONG_SP_VERSION = 1,    // "Wrong version of
stored procs on database server"
                        ERR_INVALID_CUST,                       // "Invalid
Customer id,name."
                        ERR_NO_SUCH_ORDER,                      // "No orders
found for customer."
                        ERR_RETRIED_TRANS,                      // "Retries
before transaction succeeded."
                };

                CTPCC_ODBC_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0; };

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

                int             m_errno;
                int             m_iTryCount;

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

                char *ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
        private:
                // declare variables and private functions here...
                BOOL            m_bDeadlock;                // transaction
was selected as deadlock victim
                int                             m_MaxRetries;               //
retry count on deadlock

                SQLHENV         m_henv;                                     //
ODBC environment handle
                SQLHDBC         m_hdbc;
                SQLHSTMT  m_hstmt;                      // the current hstmt

                SQLHSTMT  m_hstmtNewOrder;
                SQLHSTMT  m_hstmtPayment;
                SQLHSTMT  m_hstmtDelivery;
                SQLHSTMT  m_hstmtOrderStatus;
                SQLHSTMT  m_hstmtStockLevel;

                SQLHDESC  m_descNewOrderCols1;
                SQLHDESC  m_descNewOrderCols2;
                SQLHDESC  m_descOrderStatusCols1;
                SQLHDESC  m_descOrderStatusCols2;

                // new-order specific fields
                SQLUINTEGER     m_BindOffset;
                SQLUINTEGER     m_RowsFetched;
                int                             m_no_commit_flag;

#ifdef new_order_strstr
                // for new-order txn;
                // output params
                char            m_ol_i_name[I_NAME_LEN+1];
                double          m_ol_i_price;
                double          m_ol_amount;
                short           m_ol_stock;
                // used locally, but not returned to caller
                char            m_i_data[I_DATA_LEN];
                char            m_s_data[S_DATA_LEN];
#endif

                void ThrowError( CODBCERR::ACTION eAction );

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

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

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

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

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

};


// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

trans.h

```
/*        FILE:             TRANS.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 structure templates.
 *
 *  Change history:
 *                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
         short                                 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
         short               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
         short                                  w_id;
         short                                  d_id;
         long                                   c_id;
         short                                  c_d_id;
         short                                  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;
        short                                  ol_supply_w_id;
        short                                  ol_quantity;
        double                                 ol_amount;
        TIMESTAMP_STRUCT   ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
{       // input params
        short               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
        short                       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
        short              w_id;                       //delivery warehouse
        short              o_carrier_id;      //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{       // input params
        short                                   w_id;
        short                                   d_id;
        short                                   threshold;

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

txn_base.h

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

#pragma once

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

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

                virtual PNEW_ORDER_DATA                BuffAddr_NewOrder()
        = 0;
                virtual PPAYMENT_DATA                  BuffAddr_Payment()
        = 0;
```

```
                virtual PDELIVERY_DATA                BuffAddr_Delivery()
        = 0;
                virtual PSTOCK_LEVEL_DATA    BuffAddr_StockLevel()        = 0;
                virtual PORDER_STATUS_DATA   BuffAddr_OrderStatus()       = 0;

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

txnlog.h

```
/*      FILE:             TXNLOG.H
 *                                Microsoft TPC-C Kit Ver. 4.10.000
 *                                not yet audited
 *
 *      PURPOSE:  Header file for txn log class
 *                                Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 */

#pragma once

    typedef struct _TXN_NEWORDER
        {
        BYTE       OL_Count;           //range 0 to 31
        BYTE       OL_Remote_Count;    //range 0 to 31
        WORD       c_id;
        int                o_id;
        } TXN_NEWORDER;

    typedef struct _TXN_PAYMENT
        {
        BYTE       CustByName;
        BYTE       IsRemote;
        } TXN_PAYMENT;

    typedef struct _TXN_ORDERSTATUS
        {
                BYTE     CustByName;
        } TXN_ORDERSTATUS;

        typedef union _TXN_DETAILS
        {
                TXN_NEWORDER       NewOrder;
                TXN_PAYMENT                Payment;
                TXN_ORDERSTATUS    OrderStatus;
        } TXN_DETAILS;


        // Common header for all records in txn log.  The TxnType field is
        // a switch which identifies the particular variant.
        #define TXN_REC_TYPE_CONTROL          1        //
        #define TXN_REC_TYPE_TPCC                      2        // replaces
TRANSACTION_TYPE_TPCC
        #define TXN_REC_TYPE_TPCC_DELIV_DEF   3

        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 CTXNLOG_ERRS
                {
                        ERR_BAD_FILE_FORMAT,                    // "File
format is invalid."
                        ERR_UNKNOWN_LOG_VERSION,     // "Log file version is
unknown."
                        ERR_BROKEN_LOG_FILE,                    // "Log file
is broken."
                        ERR_LOG_NOT_SORTED,                     // "Log file
is not sorted"
                        ERR_INVALID_TIME_SEQ,                   // "Internal
Error: Record Time Sequence invalid."
                };

                CTXNLOG_ERR(int iErr) : CBaseErr(iErr) {};

                int ErrorType() {return ERR_TYPE_TXNLOG;};

                char *ErrorText()
                {
                        static char *szMsgs[] = {
                                "File format is invalid.",
                                "Log file version is unknown.",
                                "Log file is broken.",
                                "Log file is not sorted",
                                "Internal Error: Record Time Sequence
invalid.",
                                ""
                        };

                        for(int i = 0; szMsgs[i][0]; i++)
                        {
                                if ( m_idMsg == i )
                                        break;
                        }

                        return(szMsgs[i][0] ? szMsgs[i] : ERR_UNKNOWN);
                };
};
```

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;}
};
```

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;

         RegCloseKey(hKey);

         return FALSE;
}
```

ReadRegistry.h

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

```
 */

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

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

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

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );
```

WEBCLNT.DSP

```
# Microsoft Developer Studio Project File - Name="webclnt" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 5.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=webclnt - Win32 Release
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "Webclnt.mak" CFG="webclnt - Win32 Release"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "webclnt - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "webclnt - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "webclnt - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir ".\Release"
# PROP BASE Intermediate_Dir ".\Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\Release"
# PROP Intermediate_Dir ".\Release"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386

!ELSEIF  "$(CFG)" == "webclnt - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir ".\Debug"
# PROP BASE Intermediate_Dir ".\Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\Debug"
# PROP Intermediate_Dir ".\Debug"
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/c
# ADD CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386

!ENDIF

# Begin Target

# Name "webclnt - Win32 Release"
# Name "webclnt - Win32 Debug"
```

```
# End Target
# End Project
```

WebcInt.dsw

```
Microsoft Developer Studio Workspace File, Format Version 6.00
# WARNING: DO NOT EDIT OR DELETE THIS WORKSPACE FILE!

###############################################################################

Project: "db_dblib_dll"=.\db_dblib_dll\db_dblib_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "db_odbc_dll"=.\db_odbc_dll\db_odbc_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "install"=.\install\install.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name isapi_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tuxapp
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

###############################################################################

Project: "isapi_dll"=.\isapi_dll\isapi_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_tuxedo_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_com_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tm_encina_dll
    End Project Dependency
}}}

###############################################################################

Project: "tm_com_dll"=.\tm_com_dll\tm_com_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name tpcc_com_all
    End Project Dependency
}}}

###############################################################################

Project: "tm_encina_dll"=.\tm_encina_dll\tm_encina_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}
```

```
###############################################################################

Project: "tm_tuxedo_dll"=.\tm_tuxedo_dll\tm_tuxedo_dll.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tpcc_com_all"=.\tpcc_com_all\tpcc_com_all.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name tpcc_com_ps
    End Project Dependency
}}}

###############################################################################

Project: "tpcc_com_ps"=.\tpcc_com_ps\tpcc_com_ps.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
}}}

###############################################################################

Project: "tuxapp"=.\tuxapp\tuxapp.dsp - Package Owner=<4>

Package=<5>
{{{
}}}

Package=<4>
{{{
    Begin Project Dependency
    Project_Dep_Name db_dblib_dll
    End Project Dependency
    Begin Project Dependency
    Project_Dep_Name db_odbc_dll
    End Project Dependency
}}}

###############################################################################

Global:

Package=<5>
{{{
}}}

Package=<3>
{{{
}}}

###############################################################################
```

com_all_resource.h

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

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        202
#define _APS_NEXT_COMMAND_VALUE         32768
#define _APS_NEXT_CONTROL_VALUE         201
#define _APS_NEXT_SYMED_VALUE           106
#endif
#endif
```

db_dblib_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_dblib_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_dblib_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_dblib_dll.mak" CFG="db_dblib_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_dblib_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "db_dblib_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_dblib_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
```

```
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "db_dblib_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_dblib.dll"

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept

!ELSEIF  "$(CFG)" == "db_dblib_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_dblib"
# PROP BASE Intermediate_Dir "db_dblib"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept
# ADD LINK32 icap.lib ntwdblib.lib kernel32.lib user32.lib gdi32.lib winspool.lib
comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib /nologo
/subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_dblib.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_dblib_dll - Win32 Release"
# Name "db_dblib_dll - Win32 Debug"
# Name "db_dblib_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_dblib.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File
```

```
SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

db_odbc_dll.dsp

```
# Microsoft Developer Studio Project File - Name="db_odbc_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=db_odbc_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "db_odbc_dll.mak" CFG="db_odbc_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "db_odbc_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "db_odbc_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "db_odbc_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_odbc.dll"

!ELSEIF  "$(CFG)" == "db_odbc_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept

!ELSEIF  "$(CFG)" == "db_odbc_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "db_odbc_"
# PROP BASE Intermediate_Dir "db_odbc_"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
```

```
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /Gm /GX /Zi /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /D
"ICECAP" /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o /win32 "NUL"
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept
# ADD LINK32 icap.lib kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_odbc.dll"
/pdbtype:sept

!ENDIF

# Begin Target

# Name "db_odbc_dll - Win32 Release"
# Name "db_odbc_dll - Win32 Debug"
# Name "db_odbc_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp"
# Begin Source File

SOURCE=.\src\tpcc_odbc.cpp
# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

dlldata.c

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

        DO NOT ALTER THIS FILE

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

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

*********************************************************/


#include <rpcproxy.h>

#ifdef __cplusplus
extern "C"   {
#endif

EXTERN_PROXY_FILE( tpcc_com_ps )


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


DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

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

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

error.h

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

#pragma once

#ifndef _INC_STRING
```

```
        #include <string.h>
#endif

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

//error message structure used in ErrorText routines
typedef struct _SERRORMSG
{
        int             iError;                         //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_INS_MEMORY                  "Insufficient Memory to continue."
#define ERR_UNKNOWN                     "Unknown error."
#define ERR_MSG_BUF_SIZE        512
#define INV_ERROR_CODE                  -1

class CBaseErr
{
public:
        CBaseErr(LPCTSTR szLoc = NULL)
        {
                m_idMsg         = INV_ERROR_CODE;

                if (szLoc)
                {
                        m_szLoc = new char[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[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;
};

class CSocketErr : public CBaseErr
{
public:
        enum Action
        {
                eNone,
                eSend,
                eSocket,
                eBind,
                eConnect,
                eListen,
                eHost,
                eRecv,
        };

        CSocketErr(Action eAction, LPCTSTR szLocation = NULL);
        Action    m_eAction;

        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,
        };

                        CSystemErr(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;}
};
```

install.c

```
/*      FILE:             INSTALL.C
 *                                Microsoft TPC-C Kit Ver. 4.20.000
 *                                Copyright Microsoft, 1999
 *            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
 */

#include <windows.h>
```

```
#include <direct.h>
#include <io.h>
#include <stdlib.h>
#include <stdio.h>
#include <commctrl.h>
#include "..\..\common\src\ReadRegistry.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                 iThreadTimeout;
static    int                 iListenBackLog;
static    int                 iAcceptExOutstanding;

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);
static    void                ReadRegistrySettings(void);
static    void                WriteRegistrySettings(char *szDllPath);
static    BOOL                RegisterDLL(char *szFileName);
static    int                           CopyFiles(HWND hDlg, char *szDllPath);
static    BOOL                GetInstallPath(char *szDllPath);
static    void                GetVersionInfo(char *szDLLPath, char *szExePath);
static    BOOL                CheckWWWebService(void);
static    BOOL                StartWWWebService(void);
static    BOOL                StopWWWebService(void);
static    void                UpdateDialog(HWND hDlg);

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             szExePath[256];

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

                        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();

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

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


        // write binaries to inetpub\wwwroot
        rc = CopyFiles(hDlg, szDllPath);
        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;
        }

        // 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,
"SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
        {
                size = sizeof(iPoolThreadLimit);
                if ( RegQueryValueEx(hKey, "PoolThreadLimit", 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);
        }
}

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 )
          {
                    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, 15));
                    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)
{
          BOOL               bSvcRunning;

          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);
        }

        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 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_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);

        //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();
        }

        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;
        char      *ptr;
        int               iRc;

        szDllPath[0] =  0;
        bRc = TRUE;
        if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\Virtual Roots", 0,
KEY_ALL_ACCESS, &hKey) == ERROR_SUCCESS )
        {
                sv = sizeof(szData);
                iRc = RegQueryValueEx( hKey, "/,", NULL, NULL, szData, &sv );
// used by IIS 3.0
                if (iRc == ERROR_FILE_NOT_FOUND)
                        iRc = RegQueryValueEx( hKey, "/", NULL, NULL, szData,
&sv );   // used by IIS 4.0
                if (iRc == ERROR_SUCCESS)
                {
                        bRc = FALSE;
                        strcpy(szDllPath, szData);
                        if ( (ptr = strchr(szDllPath, ',')) )
                                *ptr = 0;

                        len = strlen(szDllPath);
                        if ( szDllPath[len-1] != '\\' )
                        {
                                szDllPath[len] = '\\';
                                szDllPath[len+1] = 0;
```

```
                        }
                }

                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);
        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;
}
```

install.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1                             101
#define IDI_ICON1                               102
#define IDR_TPCCDLL                             103
#define IDD_DIALOG2                             105
#define IDI_ICON2                               106
#define IDR_DELIVERY                            107
#define IDD_DIALOG3                             108

#define BN_LOG                                  1001
#define ED_KEEP                                 1002
#define ED_THREADS                              1003
#define ED_THREADS2                             1004
#define IDC_PATH                                1007
#define IDC_VERSION                             1009
#define IDC_RESULTS                             1010
#define IDC_PROGRESS1                           1011
#define IDC_STATUS                              1012
#define IDC_BUTTON1                             1013
#define ED_MAXCONNECTION                        1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT             1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE       1017
#define ED_IIS_THREAD_TIMEOUT                   1018
#define ED_IIS_LISTEN_BACKLOG                   1019
#define IDC_DBLIB                               1021
#define IDC_ODBC                                1022
#define IDC_CONNECT_POOL                                        1023
#define ED_USER_CONNECT_DELAY_TIME                      1024

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

install.rc

```
//Microsoft 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

////////////////////////////////////////////////////////////////////////////
//
// Dialog
//
```

```
IDD_DIALOG1 DIALOGEX 0, 0, 219, 351
STYLE DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP | WS_CAPTION |
    WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif"
BEGIN
    EDITTEXT        ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    CONTROL         "None",IDC_TM_NONE,"Button",BS_AUTORADIOBUTTON |
                    WS_GROUP | WS_TABSTOP,43,100,33,10
    CONTROL         "COM",IDC_TM_MTS,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,43,113,32,10
    CONTROL         "TUXEDO",IDC_TM_TUXEDO,"Button",BS_AUTORADIOBUTTON |
                    WS_TABSTOP,106,100,46,10
    CONTROL         "ENCINA",IDC_TM_ENCINA,"Button",BS_AUTORADIOBUTTON |
                    WS_DISABLED | WS_TABSTOP,106,113,43,10
    EDITTEXT        ED_DB_SERVER,131,152,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_USER_ID,131,165,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_PASSWORD,131,178,67,12,ES_AUTOHSCROLL
    EDITTEXT        ED_DB_NAME,131,191,67,12,ES_AUTOHSCROLL
    CONTROL         "DBLIB",IDC_DBLIB,"Button",BS_AUTORADIOBUTTON | WS_GROUP |
                    WS_TABSTOP,45,219,39,12
    CONTROL         "ODBC",IDC_ODBC,"Button",BS_AUTORADIOBUTTON | WS_TABSTOP,
                    91,219,39,12
    EDITTEXT        ED_IIS_MAX_THEAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
                    ES_NUMBER,WS_EX_RTLREADING
    EDITTEXT        ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    EDITTEXT        ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT | ES_NUMBER,
                    WS_EX_RTLREADING
    DEFPUSHBUTTON   "OK",IDOK,53,331,50,14
    PUSHBUTTON      "Cancel",IDCANCEL,119,331,50,14
    EDITTEXT        IDC_PATH,106,26,91,13,ES_AUTOHSCROLL | ES_READONLY
    LTEXT           "Number of Delivery Threads:",IDC_STATIC,35,45,115,12
    LTEXT           "Max Number of Connections:",IDC_STATIC,35,73,115,12
    RTEXT           "Version 4.11",IDC_VERSION,120,4,89,9
    LTEXT           "IIS Max Thread Pool Limit:",IDC_STATIC,36,263,115,12
    LTEXT           "Web Service Backlog Queue Size:",IDC_STATIC,36,277,115,
                    12
    LTEXT           "IIS Thread Timeout (seconds):",IDC_STATIC,36,291,115,12
    LTEXT           "IIS Listen Backlog:",IDC_STATIC,36,307,115,10
    GROUPBOX        "Database Interface",IDC_STATIC,35,208,163,27,WS_GROUP
    LTEXT           "Installation directory:",IDC_STATIC,35,29,71,10
    GROUPBOX        "Transaction Monitor",IDC_STATIC,33,90,165,37
    LTEXT           "Server Name:",IDC_STATIC,35,155,56,8
    LTEXT           "User ID:",IDC_STATIC,35,168,60,8
    LTEXT           "User Password:",IDC_STATIC,35,181,83,8
    LTEXT           "Database Name:",IDC_STATIC,35,194,54,8
    GROUPBOX        "SQL Server Connection Properties",IDC_STATIC,22,139,187,
                    102
    GROUPBOX        "Web Client Properties",IDC_STATIC,22,15,187,118
    GROUPBOX        "IIS Settings",IDC_STATIC,22,247,187,79
    LTEXT           "Max Pending Deliveries:",IDC_STATIC,35,59,115,12
END

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

    IDD_DIALOG3 DIALOG DISCARDABLE  0, 0, 91, 40
    STYLE DS_SYSMODAL | DS_MODALFRAME | DS_3DLOOK | DS_CENTER | WS_CAPTION
    CAPTION "Installing TPC-C Web Client"
    FONT 12, "Arial Black"
    BEGIN
        CONTROL         "Progress1",IDC_PROGRESS1,"msctls_progress32",WS_BORDER,
                        7,20,77,13
        CTEXT           "Static",IDC_STATUS,7,7,77,12,SS_SUNKEN
    END

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


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

    #ifdef APSTUDIO_INVOKED
    GUIDELINES DESIGNINFO DISCARDABLE
    BEGIN
        IDD_DIALOG1, DIALOG
        BEGIN
            LEFTMARGIN, 22
            RIGHTMARGIN, 209
            VERTGUIDE, 35
            VERTGUIDE, 198
            TOPMARGIN, 4
            BOTTOMMARGIN, 345
        END

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

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

```
        BOTTOMMARGIN, 33
    END

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


#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


//////////////////////////////////////////////////////////////////////////////
//
// Icon
//

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

//////////////////////////////////////////////////////////////////////////////
//
// TPCCDLL
//

IDR_TPCCDLL             TPCCDLL DISCARDABLE     "..\\..\\isapi_dll\\bin\\tpcc.dll"

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

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

#endif    // !_MAC


//////////////////////////////////////////////////////////////////////////////
//
// LICENSE
//

IDR_LICENSE1            LICENSE DISCARDABLE     "license.txt"

//////////////////////////////////////////////////////////////////////////////
//
// DBLIB_DLL
//

IDR_DBLIB_DLL           DBLIB_DLL DISCARDABLE
"..\\..\\db_dblib_dll\\bin\\tpcc_dblib.dll"

//////////////////////////////////////////////////////////////////////////////
//
// ODBC_DLL
//

IDR_ODBC_DLL            ODBC_DLL DISCARDABLE
"..\\..\\db_odbc_dll\\bin\\tpcc_odbc.dll"

//////////////////////////////////////////////////////////////////////////////
//
// TUXEDO_APP
```

```
//

IDR_TUXEDO_APP          TUXEDO_APP DISCARDABLE  "..\\..\\tuxapp\\bin\\tuxapp.exe"

/////////////////////////////////////////////////////////////////////////////
//
// TUXEDO_DLL
//

IDR_TUXEDO_DLL          TUXEDO_DLL DISCARDABLE
"..\\..\\tm_tuxedo_dll\\bin\\tpcc_tuxedo.dll"

/////////////////////////////////////////////////////////////////////////////
//
// COM_DLL
//

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

/////////////////////////////////////////////////////////////////////////////
//
// COM_PS_DLL
//

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

/////////////////////////////////////////////////////////////////////////////
//
// COM_ALL_DLL
//

IDR_COMALL_DLL          COM_ALL_DLL DISCARDABLE
"..\\..\\tpcc_com_all\\bin\\tpcc_com_all.dll"
#endif    // English (U.S.) resources
/////////////////////////////////////////////////////////////////////////////


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


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

install_com.cpp

```
/*        FILE:             INSTALL_COM.CPP
 *                                  Microsoft TPC-C Kit Ver. 4.20.000
 *                                  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 =        "";
        // 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;
}
```

install_resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Developer Studio generated include file.
// Used by install.rc
//
#define IDD_DIALOG1                     101
#define IDI_ICON1                       102
#define IDR_TPCCDLL                     103
#define IDD_DIALOG2                     105
#define IDI_ICON2                       106
#define IDR_DELIVERY                    107
#define IDD_DIALOG3                     108
#define IDR_LICENSE1                    112
#define IDD_DIALOG4                     113
#define IDR_TPCCOBJ1                    117
#define IDR_TPCCSTUB1                   118
#define IDR_DBLIB_DLL                   122
#define IDR_ODBC_DLL                    123
#define IDR_TUXEDO_APP                  124
#define IDR_TUXEDO_DLL                  125
#define IDR_COM_DLL                     126
#define IDR_COMPS_DLL                   127
#define IDR_COMALL_DLL                  128
#define BN_LOG                          1001
#define ED_KEEP                         1002
#define ED_THREADS                      1003
#define ED_THREADS2                     1004
#define IDC_PATH                        1007
#define IDC_VERSION                     1009
#define IDC_RESULTS                     1010
#define IDC_PROGRESS1                   1011
#define IDC_STATUS                      1012
#define IDC_BUTTON1                     1013
#define ED_MAXCONNECTION                1014
#define ED_IIS_MAX_THEAD_POOL_LIMIT     1015
#define ED_MAXDELIVERIES                1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT           1018
#define ED_IIS_LISTEN_BACKLOG           1019
#define IDC_DBLIB                       1021
#define IDC_LICENSE                     1022
#define IDC_ODBC                        1022
#define IDC_CONNECT_POOL                1023
#define ED_DB_SERVER                    1023
#define ED_USER_CONNECT_DELAY_TIME      1024
#define ED_DB_USER_ID                   1024
#define IDC_MTS                         1025
#define IDC_TM_MTS                      1025
#define IDC_TM_TUXEDO                   1026
#define IDC_TM_NONE                     1027
#define ED_DB_PASSWORD                  1028
#define ED_DB_NAME                      1029
#define IDC_TM_ENCINA                   1030

// Next default values for new objects
//
#ifdef APSTUDIO_INVOKED
#ifndef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE        129
#define _APS_NEXT_COMMAND_VALUE         40001
#define _APS_NEXT_CONTROL_VALUE         1024
#define _APS_NEXT_SYMED_VALUE           101
#endif
#endif
```

isapi_dll.dsp

```
# Microsoft Developer Studio Project File - Name="isapi_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=isapi_dll - Win32 IceCAP
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "isapi_dll.mak" CFG="isapi_dll - Win32 IceCAP"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "isapi_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "isapi_dll - Win32 IceCAP" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "isapi_dll - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
```

```
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "NDEBUG" /D "WIN32" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /machine:I386
/nodefaultlib:"LIBCMT" /out:".\bin/tpcc.dll"
# SUBTRACT LINK32 /nodefaultlib

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /GX /ZI /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 ..\common\txnlog\lib\debug\rtetime.lib
..\common\txnlog\lib\debug\spinlock.lib ..\common\txnlog\lib\debug\error.lib
..\common\txnlog\lib\debug\txnlog.lib wsock32.lib kernel32.lib user32.lib gdi32.lib
winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib
odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug /machine:I386
/nodefaultlib:"LIBCMTD" /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /nodefaultlib

!ELSEIF  "$(CFG)" == "isapi_dll - Win32 IceCAP"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "isapi_dl"
# PROP BASE Intermediate_Dir "isapi_dl"
# PROP BASE Ignore_Export_Lib 0
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MDd /W3 /GX /Zi /Od /D "_DEBUG" /D "WIN32" /D "_WINDOWS" /FR
/YX /FD /Gh /c
# ADD CPP /nologo /MD /W3 /GX /Zi /O2 /D "NDEBUG" /D "ICECAP" /D "WIN32" /D
"_WINDOWS" /FR /YX /FD /Gh /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc.dll"
/pdbtype:sept
# SUBTRACT BASE LINK32 /profile /pdb:none
# ADD LINK32 icap.lib ..\common\txnlog\lib\release\rtetime.lib
..\common\txnlog\lib\release\spinlock.lib ..\common\txnlog\lib\release\error.lib
..\common\txnlog\lib\release\txnlog.lib wsock32.lib kernel32.lib user32.lib
gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib ole32.lib oleaut32.lib
uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows /dll /debug
/machine:I386 /out:".\bin/tpcc.dll" /pdbtype:sept
# SUBTRACT LINK32 /profile /pdb:none /map

!ENDIF

# Begin Target

# Name "isapi_dll - Win32 Release"
# Name "isapi_dll - Win32 Debug"
# Name "isapi_dll - Win32 IceCAP"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.def, *.rc"
# Begin Source File

SOURCE=.\src\tpcc.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.rc
# End Source File
# End Group
# Begin Group "Header Files"

# PROP Default_Filter "*.h, *.hpp"
# Begin Source File
```

```
SOURCE=..\common\src\error.h
# End Source File
# Begin Source File

SOURCE=..\common\src\ReadRegistry.h
# End Source File
# Begin Source File

SOURCE=.\src\tpcc.h
# End Source File
# Begin Source File

SOURCE=..\db_dblib_dll\src\tpcc_dblib.h
# End Source File
# Begin Source File

SOURCE=..\db_odbc_dll\src\tpcc_odbc.h
# End Source File
# Begin Source File

SOURCE=..\tm_tuxedo_dll\src\tpcc_tux.h
# End Source File
# Begin Source File

SOURCE=..\common\src\trans.h
# End Source File
# Begin Source File

SOURCE=..\common\src\txn_base.h
# End Source File
# End Group
# End Target
# End Project
```

isapi_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
```

rtetime.h

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

//FILE: RTETIME.H

#define   MAX_JULIAN_TIME             0x7FFFFFFFFFFFFFFF
#define JULIAN_TIME__int64
#define   TC_TIME             DWORD
extern "C"
{
BOOL                InitJulianTime(LPSYSTEMTIME lpInitTime);
JULIAN_TIME         GetJulianTime(void);
DWORD               MyTickCount(void);
void                GetJulianAndTC(JULIAN_TIME *pJulian, DWORD *pTC);
JULIAN_TIME         ConvertTo64BitTime(int iYear, int iMonth, int iDay, int iHour,
int iMinute, int iSecond);
JULIAN_TIME         Get64BitTime(LPSYSTEMTIME lpInitTime);
int                         JulianDay( int yr, int mm, int dd );
void                JulianToTime(JULIAN_TIME julianTS, int* yr, int* mm, int* dd,
int *hh, int *mi, int *ss );
void                JulianToCalendar( int day, int* yr, int* mm, int* dd );
}
```

spinlock.h

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

#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
```

tm_com_dll.dsp

```
# Microsoft Developer Studio Project File - Name="tm_com_dll" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tm_com_dll - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tm_com_dll.mak" CFG="tm_com_dll - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tm_com_dll - Win32 Release" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE "tm_com_dll - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tm_com_dll - Win32 Release"
```

```
# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MD /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386 /out:".\bin/tpcc_com.dll"

!ELSEIF  "$(CFG)" == "tm_com_dll - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MDd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /out:".\bin/tpcc_com.dll"
/pdbtype:sept
!ENDIF

# Begin Target

# Name "tm_com_dll - Win32 Release"
# Name "tm_com_dll - Win32 Debug"
# Begin Source File

SOURCE=.\src\tpcc_com.cpp
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com.h
# End Source File
# End Target
# End Project
```

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];

         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

        int                                 iRetryCnt = 0;
        static int                  iMaxRetries = 10;

        assert(txnDelilog != NULL);

Reconnect:
        try
        {
                if (Reg.eDB_Protocol == ODBC)
                        pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                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;

                // will retry connection up to ten times
                if (iRetryCnt++ < iMaxRetries)
                {
                        Sleep(5000);       // delay for 5 seconds
                        goto Reconnect;
                }

                wsprintf( szTmp, "Delivery Txn thread terminating after %d
retries.", iMaxRetries );
                WriteMessageToEventLog( szTmp );
                goto ErrorExit;
        }
        catch (...)
        {
                WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread.  Delivery Txn thread terminating."));
                goto ErrorExit;
        }


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

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

                                // make a local copy of current entry from
delivery buffer and increment buffer index

        EnterCriticalSection(&DelBuffCriticalSection);
                                delivery = *(pDelBuff+dwDelBuffBusyIndex);
                                dwDelBuffFreeCount++;
                                dwDelBuffBusyIndex++;
                                if (dwDelBuffBusyIndex == dwDelBuffSize)
        // wrap-around if at end of buffer
                                        dwDelBuffBusyIndex = 0;


        LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

```
                                        GetLocalTime( &trans_end );

                                        //log txn
                                        txnDeliRec.TxnStatus = ERR_SUCCESS;
                                        for (int i=0; i<10; i++)
                                                txnDeliRec.o_id[i] =
pDeliveryData->o_id[i];
                                        txnDeliRec.DeltaT4 =
(int)(Get64BitTime(&trans_end) - txnDeliRec.TxnStartT0);
                                        txnDeliRec.DeltaTxnExec =
(int)(Get64BitTime(&trans_end) - Get64BitTime(&trans_start));

                                        if (txnDelilog != NULL)
                                                txnDelilog-
>WriteToLog(&txnDeliRec);
                                }
                        }
                        catch (CBaseErr *e)
                        {
                                char szTmp[1024];
                                wsprintf( szTmp, "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(short 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=4><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 );
                else if (Reg.eDB_Protocol == DBLIB)
                        Term.pClientData[iNewTerm].pTxn = pCTPCC_DBLIB_new(
szServer, szUser, szPassword, szMyComputerName, szDatabase );
        }
        catch (...)
        {
                TermDelete(iNewTerm);
                throw;              // pass exception upward
        }

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

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

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

        EnterCriticalSection(&TermCriticalSection);

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

        LeaveCriticalSection(&TermCriticalSection);

        wsprintf( szBuffer,
                        "<HTML><HEAD><TITLE>TPC-C Web Client
Stats</TITLE></HEAD>"
```

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

char *CWEBCLNT_ERR::ErrorText()
{
        static SERRORMSG errorMsgs[] =
        {
                {       ERR_COMMAND_UNDEFINED,
                "Command undefined."                                            },

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

                {       ERR_LOADDLL_FAILED,
                "Load of DLL failed.  DLL="
                                                                },
                {       ERR_MAX_CONNECTIONS_EXCEEDED,                   "No
connections available.  Max Connections is probably too low."   },
                {       ERR_MISSING_REGISTRY_ENTRIES,
        "Required registry entries are missing.  Rerun INSTALL to correct."},
                {       ERR_NEWORDER_CUSTOMER_INVALID,
        "New Order customer id invalid data type, range = 1 to 3000."
        },
                {       ERR_NEWORDER_CUSTOMER_KEY,
        "New Order missing Customer key \"CID*\"."
                                },
                {       ERR_NEWORDER_DISTRICT_INVALID,
        "New Order District ID Invalid range 1 - 10."
                        },
                {       ERR_NEWORDER_FORM_MISSING_DID,
        "New Order missing District key \"DID*\"."
                                },
                {       ERR_NEWORDER_ITEMID_INVALID,                    "New
Order Item Id is wrong data type, must be numeric."                     },
                {       ERR_NEWORDER_ITEMID_RANGE,
        "New Order Item Id is out of range. Range = 1 to 999999."
        },
                {       ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,              "New
Order Item_Id field entered without a corresponding Supp_W."    },
                {       ERR_NEWORDER_MISSING_IID_KEY,                   "New
Order missing Item Id key \"IID*\"."
                },
                {       ERR_NEWORDER_MISSING_QTY_KEY,                   "New
Order Missing Qty key \"Qty##*\"."
                        },
                {       ERR_NEWORDER_MISSING_SUPPW_KEY,
        "New Order missing Supp_W key \"SP##*\"."
                                },
                {       ERR_NEWORDER_NOITEMS_ENTERED,                   "New
Order No order lines entered."
                                        },
                {       ERR_NEWORDER_QTY_INVALID,
        "New Order Qty invalid must be numeric range 1 - 99."
                },
                {       ERR_NEWORDER_QTY_RANGE,
                "New Order Qty is out of range. Range = 1 to 99."
                },
                {       ERR_NEWORDER_QTY_WITHOUT_SUPPW,
        "New Order Qty field entered without a corresponding Supp_W."
        },
                {       ERR_NEWORDER_SUPPW_INVALID,
        "New Order Supp_W invalid data type must be numeric."
                },
                {       ERR_NO_SERVER_SPECIFIED,
        "No Server name specified."
                                                },
                {       ERR_ORDERSTATUS_CID_AND_CLT,
        "Order Status Only Customer ID or Last Name may be entered, not both."
        },
                {       ERR_ORDERSTATUS_CID_INVALID,
        "Order Status Customer ID invalid, range must be numeric 1 - 3000."},
                {       ERR_ORDERSTATUS_CLT_RANGE,
        "Order Status Customer last name longer than 16 characters."
        },
                {       ERR_ORDERSTATUS_DID_INVALID,
        "Order Status District invalid, value must be numeric 1 - 10."
        },
                {       ERR_ORDERSTATUS_MISSING_CID_CLT,
        "Order Status Either Customer ID or Last Name must be entered."
        },
                {       ERR_ORDERSTATUS_MISSING_CID_KEY,
        "Order Status missing Customer key \"CID*\"."
                        },
                {       ERR_ORDERSTATUS_MISSING_CLT_KEY,
        "Order Status missing Customer Last Name key \"CLT*\"."
                },
                {       ERR_ORDERSTATUS_MISSING_DID_KEY,
        "Order Status missing District key \"DID*\"."
                        },
                {       ERR_PAYMENT_CDI_INVALID,
        "Payment Customer district invalid must be numeric."
                },
                {       ERR_PAYMENT_CID_AND_CLT,
        "Payment Only Customer ID or Last Name may be entered, not both."  },
                {       ERR_PAYMENT_CUSTOMER_INVALID,
        "Payment Customer data type invalid, must be numeric."
                },
                {       ERR_PAYMENT_CWI_INVALID,
        "Payment Customer Warehouse invalid, must be numeric."
                },
```

```
                {       ERR_PAYMENT_DISTRICT_INVALID,
        "Payment District ID is invalid, must be 1 - 10."
                },
                {       ERR_PAYMENT_HAM_INVALID,
        "Payment Amount invalid data type must be numeric."
                        },
                {       ERR_PAYMENT_HAM_RANGE,
                "Payment Amount out of range, 0 - 9999.99."
                                        },
                {       ERR_PAYMENT_LAST_NAME_TO_LONG,
        "Payment Customer last name longer than 16 characters."
                },
                {       ERR_PAYMENT_MISSING_CDI_KEY,
        "Payment missing Customer district key \"CDI*\"."
                },
                {       ERR_PAYMENT_MISSING_CID_CLT,
        "Payment Either Customer ID or Last Name must be entered."
        },
                {       ERR_PAYMENT_MISSING_CID_KEY,
        "Payment missing Customer Key \"CID*\"."
                                },
                {       ERR_PAYMENT_MISSING_CLT_KEY,
        "Payment missing Customer Last Name key \"CLT*\"."
                        },
                {       ERR_PAYMENT_MISSING_CWI_KEY,
        "Payment missing Customer Warehouse key \"CWI*\"."
                        },
                {       ERR_PAYMENT_MISSING_DID_KEY,
        "Payment missing District Key \"DID*\"."
                                },
                {       ERR_PAYMENT_MISSING_HAM_KEY,
        "Payment missing Amount key \"HAM*\"."
                                },
                {       ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,  "Stock Level;
missing Threshold key \"TT*\"."
        },
                {       ERR_STOCKLEVEL_THRESHOLD_INVALID,
        "Stock Level; Threshold value must be in the range = 1 - 99."
        },
                {       ERR_STOCKLEVEL_THRESHOLD_RANGE,
        "Stock Level Threshold out of range, range must be 1 - 99."
        },
                {       ERR_VERSION_MISMATCH,
                "Invalid version field.  RTE and Web Client are probably out of
sync." },
                {       ERR_W_ID_INVALID,
                "Invalid Warehouse ID."
                                                                        },
                {       0,
                                        ""

                                                        }
        };

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

        if (m_szTextDetail)
                strcat( szTmp, m_szTextDetail );
        if (m_SystemErr)
                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: %4.4d   District: %2.2d<BR> <BR>",
                    STOCK_LEVEL_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
                    Term.pClientData[iTermId].w_id, Term.pClientData[iTermId].d_id);

          if ( bInput )
          {
                    strcpy(szForm+c,
                              "Stock Level Threshold: <INPUT NAME=\"TT*\"
SIZE=2><BR> <BR>"
                              "low stock:    </font><BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR> <BR>"
                              " <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"
                              "<INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"Process\">"
                              "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
                              "</FORM></HTML>" );
          }
          else
          {
                    wsprintf(szForm+c,
                              "Stock Level Threshold: %2.2d<BR> <BR>"
                              "low stock: %3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR>
<BR> <BR> <BR>"
                              " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></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: %4.4d   ",
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: %4.4d   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, "  %4.4d   %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: %4.4d"
                        "                          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: %4.4d
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: %4.4d  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> <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: %4.4d   ",
                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></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, "  %4.4d       %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: %4.4d<BR> <BR>",
                (!bInput && (pDeliveryData->exec_status_code != eOK)) ?
ERR_TYPE_DELIVERY_POST : 0,
                DELIVERY_FORM, iTermId, Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

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

tpcc.def

```
LIBRARY TPCC.DLL

EXPORTS

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

tpcc.h

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

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

#define TP_MAX_RETRIES
         50


//note that the welcome form must be processed first as terminal ids assigned here,
once the
```

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

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


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

         int                          iSyncId;
         //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(short w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);
```

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
```

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 );
}
```

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.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.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.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_all" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Dynamic-Link Library" 0x0102

CFG=tpcc_com_all - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_all.mak" CFG="tpcc_com_all - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_all - Win32 Release" (based on "Win32 (x86) Dynamic-Link
Library")
!MESSAGE "tpcc_com_all - Win32 Debug" (based on "Win32 (x86) Dynamic-Link Library")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD
/c
# ADD CPP /nologo /MT /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /machine:I386
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /machine:I386

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /MTd /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS"
/YX /FD /c
# ADD CPP /nologo /MTd /W3 /Gm /GX /ZI /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /dll /debug /machine:I386 /pdbtype:sept
```

```
# ADD LINK32 ..\db_dblib_dll\bin\tpcc_dblib.lib ..\db_odbc_dll\bin\tpcc_odbc.lib
kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib advapi32.lib shell32.lib
ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib /nologo /subsystem:windows
/dll /debug /machine:I386 /pdbtype:sept

!ENDIF

# Begin Target

# Name "tpcc_com_all - Win32 Release"
# Name "tpcc_com_all - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter "*.cpp, *.c"
# Begin Source File

SOURCE=.\src\tpcc_com_all.cpp
# SUBTRACT CPP /YX
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.def
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_all.idl

!IF  "$(CFG)" == "tpcc_com_all - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
         midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_all - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build - Performing MIDL step
InputPath=.\src\tpcc_com_all.idl

BuildCmds= \
         midl /Oicf /h "tpcc_com_all.h" /iid "tpcc_com_all_i.c"
".\src\tpcc_com_all.idl"   /out ".\src"

".\src\tpcc_com_all.tlb" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_all_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# End Group
# Begin Group "Header"

# PROP Default_Filter "*.h"
# Begin Source File

SOURCE=.\src\Methods.h
# End Source File
# Begin Source File

SOURCE=.\src\resource.h
# End Source File
# End Group
# Begin Source File

SOURCE=.\src\tpcc_com_all.rc
# End Source File
# End Target
# End Project
```

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 Mon Jun 12 18:15:19 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.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.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.rgs

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

```
        }
        TPCC.AllTxns = s 'TPCC Class'
        {
                CurVer = s 'TPCC.AllTxns.1'
        }
        NoRemove CLSID
        {
                ForceRemove {122A3128-2520-11D3-BA71-00C04FBFE08B} = s 'TPCC
Class'
                {
                        ProgID = s 'TPCC.AllTxns.1'
                        VersionIndependentProgID = s 'TPCC.AllTxns'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_com_all_i.c

```
#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 Mon Jun 12 18:15:19 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 Mon Jun 12 18:15:19 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_no.rgs

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

```
}
```

tpcc_com_os.rgs

```
HKCR
{
        TPCC.OrderStatus.1 = s 'OrderStatus Class'
        {
                CLSID = s '{266836AD-A50D-11D2-BA4E-00C04FBFE08B}'
        }
        TPCC.OrderStatus = s 'OrderStatus Class'
        {
                CurVer = s 'TPCC.OrderStatus.1'
        }
        NoRemove CLSID
        {
                ForceRemove {266836AD-A50D-11D2-BA4E-00C04FBFE08B} = s
'OrderStatus Class'
                {
                        ProgID = s 'TPCC.OrderStatus.1'
                        VersionIndependentProgID = s 'TPCC.OrderStatus'
                        InprocServer32 = s '%MODULE%'
                        {
                                val ThreadingModel = s 'Both'
                        }
                }
        }
}
```

tpcc_com_pay.rgs

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

tpcc_com_ps.def

```
LIBRARY      "tpcc_com_ps"

DESCRIPTION  'Proxy/Stub DLL'

EXPORTS
        DllGetClassObject       @1   PRIVATE
        DllCanUnloadNow         @2   PRIVATE
        GetProxyDllInfo         @3   PRIVATE
        DllRegisterServer            @4        PRIVATE
        DllUnregisterServer          @5        PRIVATE
```

tpcc_com_ps.dsp

```
# Microsoft Developer Studio Project File - Name="tpcc_com_ps" - Package Owner=<4>
# Microsoft Developer Studio Generated Build File, Format Version 6.00
# ** DO NOT EDIT **

# TARGTYPE "Win32 (x86) Application" 0x0101

CFG=tpcc_com_ps - Win32 Debug
!MESSAGE This is not a valid makefile. To build this project using NMAKE,
!MESSAGE use the Export Makefile command and run
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak".
!MESSAGE
!MESSAGE You can specify a configuration when running NMAKE
!MESSAGE by defining the macro CFG on the command line. For example:
!MESSAGE
!MESSAGE NMAKE /f "tpcc_com_ps.mak" CFG="tpcc_com_ps - Win32 Debug"
!MESSAGE
!MESSAGE Possible choices for configuration are:
!MESSAGE
!MESSAGE "tpcc_com_ps - Win32 Release" (based on "Win32 (x86) Application")
!MESSAGE "tpcc_com_ps - Win32 Debug" (based on "Win32 (x86) Application")
!MESSAGE

# Begin Project
# PROP AllowPerConfigDependencies 0
# PROP Scc_ProjName ""
# PROP Scc_LocalPath ""
CPP=cl.exe
MTL=midl.exe
RSC=rc.exe

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 0
# PROP BASE Output_Dir "Release"
# PROP BASE Intermediate_Dir "Release"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 0
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D "_WINDOWS" /YX /FD /c
# ADD CPP /nologo /W3 /GX /O2 /D "WIN32" /D "NDEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# SUBTRACT CPP /YX
# ADD BASE MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
```

```
# ADD MTL /nologo /D "NDEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "NDEBUG"
# ADD RSC /l 0x409 /d "NDEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /machine:I386
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /subsystem:windows /dll /pdb:none /machine:I386
/def:".\src\tpcc_com_ps.def"
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
         copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP BASE Use_MFC 0
# PROP BASE Use_Debug_Libraries 1
# PROP BASE Output_Dir "Debug"
# PROP BASE Intermediate_Dir "Debug"
# PROP BASE Target_Dir ""
# PROP Use_MFC 0
# PROP Use_Debug_Libraries 1
# PROP Output_Dir ".\bin"
# PROP Intermediate_Dir ".\obj"
# PROP Ignore_Export_Lib 0
# PROP Target_Dir ""
# ADD BASE CPP /nologo /W3 /Gm /GX /Zi /Od /D "WIN32" /D "_DEBUG" /D "_WINDOWS" /YX
/FD /c
# ADD CPP /nologo /ZI /Od /D "WIN32" /D "_DEBUG" /D _WIN32_WINNT=0x0400 /D
"REGISTER_PROXY_DLL" /FD /c
# ADD BASE MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD MTL /nologo /D "_DEBUG" /mktyplib203 /o "NUL" /win32
# ADD BASE RSC /l 0x409 /d "_DEBUG"
# ADD RSC /l 0x409 /d "_DEBUG"
BSC32=bscmake.exe
# ADD BASE BSC32 /nologo
# ADD BSC32 /nologo
LINK32=link.exe
# ADD BASE LINK32 kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib odbc32.lib odbccp32.lib
/nologo /subsystem:windows /debug /machine:I386 /pdbtype:sept
# ADD LINK32 kernel32.lib rpcndr.lib rpcns4.lib rpcrt4.lib oleaut32.lib uuid.lib
/nologo /entry:"DllMain" /dll /debug /machine:IX86 /def:".\src\tpcc_com_ps.def"
/pdbtype:sept
# SUBTRACT LINK32 /pdb:none
# Begin Custom Build - Copying tpcc_com_ps.h
InputPath=.\bin\tpcc_com_ps.dll
SOURCE="$(InputPath)"

"..\tpcc_com_all\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
         copy    .\src\tpcc_com_ps.h    ..\tpcc_com_all\src\

# End Custom Build

!ENDIF

# Begin Target

# Name "tpcc_com_ps - Win32 Release"
# Name "tpcc_com_ps - Win32 Debug"
# Begin Group "Source"

# PROP Default_Filter ""
# Begin Source File

SOURCE=.\src\dlldata.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.def
# PROP Exclude_From_Build 1
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps.idl

!IF  "$(CFG)" == "tpcc_com_ps - Win32 Release"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
         midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ELSEIF  "$(CFG)" == "tpcc_com_ps - Win32 Debug"

# PROP Ignore_Default_Tool 1
# Begin Custom Build
InputPath=.\src\tpcc_com_ps.idl

BuildCmds= \
         midl /Oicf /h "tpcc_com_ps.h" /iid "tpcc_com_ps_i.c"
".\src\tpcc_com_ps.idl"    /out ".\src"

".\src\tpcc_com_ps.h" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\tpcc_com_ps_i.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)

".\src\dlldata.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
```

```
".\src\tpcc_com_ps_p.c" : $(SOURCE) "$(INTDIR)" "$(OUTDIR)"
   $(BuildCmds)
# End Custom Build

!ENDIF

# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_i.c
# End Source File
# Begin Source File

SOURCE=.\src\tpcc_com_ps_p.c
# End Source File
# End Group
# End Target
# End Project
```

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 Mon Jun 12 18:15:12 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.idl

```
/*       FILE:              ITPCC.IDL
 *                                  Microsoft TPC-C Kit Ver. 4.20.000
 *                                  Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 *                                  not yet audited
 *
 *       PURPOSE:  Defines the interface used by TPCC. This interface can be
implemented by C++ components.
 *
 *  Change history:
 *                4.20.000 - first version
 */

// Forward declare all types defined
interface ITPCC;
import "oaidl.idl";
import "ocidl.idl";

         [
                  object,
                  oleautomation,
                  uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),
                  helpstring("ITPCC Interface"),
                  pointer_default(unique)
         ]
         interface ITPCC : IUnknown
         {

        HRESULT _stdcall NewOrder
                                                          (
                                                          [in] VARIANT  txn_in,
                                                          [out] VARIANT *txn_out
                                                          );

        HRESULT _stdcall Payment
                                                          (
                                                          [in] VARIANT txn_in,
                                                          [out] VARIANT *txn_out
                                                          );

        HRESULT _stdcall Delivery
                                                          (
                                                          [in] VARIANT txn_in,
                                                          [out] VARIANT *txn_out
                                                          );

        HRESULT _stdcall StockLevel
                                                          (
                                                          [in] VARIANT txn_in,
                                                          [out] VARIANT *txn_out
                                                          );
```

```
        HRESULT _stdcall OrderStatus
                                                (
                                                [in] VARIANT txn_in,
                                                [out] VARIANT *txn_out
                                                );

        HRESULT _stdcall CallSetComplete
                                                (
                                                );

    };    // interface ITPCC
```

tpcc_com_ps_i.c

```
#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 Mon Jun 12 18:15:12 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 Mon Jun 12 18:15:12 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_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 Mon Jun 12 18:15:12 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,0x47,0x00,0xC0,0x4F,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,
    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 5.3.280 */
    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( 0xfffffff8 ),    /* 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( 0xffffffe8 ),    /* 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( 0xffffffe8 ),    /* 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( 0xfffffffee ),    /* Offset= -18 (836) */
/* 856 */ 0x5c,              /* FC_PAD */
                              0x5b,              /* FC_END */
/* 858 */
                              0x1a,              /* FC_BOGUS_STRUCT */
                              0x3,               /* 3 */
/* 860 */ NdrFcShort( 0x28 ),/* 40 */
/* 862 */ NdrFcShort( 0xfffffffee ),    /* 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( 0xffffffdf7 ),    /* Offset= -521 (352) */
                              0x5b,              /* FC_END */
/* 876 */
                              0x12, 0x0,         /* FC_UP */
/* 878 */ NdrFcShort( 0xffffffef6 ),    /* 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 Mon Jun 12 18:15:12 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 5.3.280 */
    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( 0xffffffdc ),    /* 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( 0xfffffff7 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffffe6 ),    /* 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( 0xffffff02 ),    /* 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( 0xfffffda4 ),    /* Offset= -604 (280) */
/* 886 */
                              0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 888 */ NdrFcShort( 0xfffffda6 ),    /* Offset= -602 (286) */
/* 890 */
                              0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 892 */ NdrFcShort( 0xfffffdbc ),    /* Offset= -580 (312) */
/* 894 */
                              0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 896 */ NdrFcShort( 0xfffffdca ),    /* Offset= -566 (330) */
/* 898 */
                              0x12, 0x10,        /* FC_UP [pointer_deref] */
/* 900 */ NdrFcShort( 0xfffffdd8 ),    /* 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( 0xffffffdc ),    /* 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)*/
```

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_dblib.cpp

```
/*        FILE:             TPCC_DBLIB.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:  Implements dblib calls for TPC-C txns.
 *        Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                  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);

        // set connection properties to match those used by ODBC
        dbcmd(m_dbproc, "set ANSI_DEFAULTS ON ");
        dbcmd(m_dbproc, "set CURSOR_CLOSE_ON_COMMIT OFF ");
        dbcmd(m_dbproc, "set IMPLICIT_TRANSACTIONS OFF ");
        dbcmd(m_dbproc, "set NOCOUNT ON ");                     // do not
return row counts
        dbcmd(m_dbproc, "set XACT_ABORT ON ");          // rollback transaction
on abort

        // for coyote
//      dbcmd(m_dbproc, "set ansi_warnings on ");       //
//      dbcmd(m_dbproc, "set ansi_nulls on ");          //

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

        // This value must match the number of commands above.
//      DiscardNextResults(2);
        DiscardNextResults(5);      // coyote

        // 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, SQLINT2, -1, -1, (BYTE
*) &m_txn.StockLevel.w_id);          // @w_id smallint
                              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, SQLINT2, -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, SQLINT2, -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, SQLINT2, -1, -1, (BYTE
*) &m_txn.Payment.w_id);
                        dbrpcparam(m_dbproc, NULL, 0, SQLINT2, -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, SQLINT2, -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, SQLINT2, -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;
}
```

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."
                        ERR_RETRIED_TRANS,                      // "Retries
before transaction succeeded."
                };

                CTPCC_DBLIB_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0;
};

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

                int               m_errno;
                int               m_iTryCount;

                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);
```

tpcc_odbc.cpp

```
/*       FILE:              TPCC_ODBC.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:  Implements ODBC calls for TPC-C txns.
 *       Contact:  Charles Levine (clevine@microsoft.com)
 *
 *  Change history:
 *                4.20.000 - updated rev number to match kit
 *                4.10.001 - not deleting error class in catch handler on deadlock
retry;
 *                                     not a functional bug, but a memory leak
 */

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

#define DBNTWIN32
#include <sqltypes.h>
#include <sql.h>
#include <sqlext.h>
#include <odbcss.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_odbc.h"

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

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

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

static SQLHENV henv = SQL_NULL_HENV;                              // ODBC
environment handle

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
         switch( ul_reason_for_call )
         {
                  case DLL_PROCESS_ATTACH:
                           DisableThreadLibraryCalls(hModule);
                           if ( SQLAllocHandleStd(SQL_HANDLE_ENV,
SQL_NULL_HANDLE, &henv) != SQL_SUCCESS )
                                    return FALSE;
                           break;

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

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

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
 *
 */

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

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

                                                    }
         };
```

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

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

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ODBC* CTPCC_ODBC_new(
        LPCSTR szServer,            // name of SQL server
        LPCSTR szUser,                      // user name for login
        LPCSTR szPassword,          // password for login
        LPCSTR szHost,                      // not used
        LPCSTR szDatabase )         // name of database to use
{
        return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase );
}


CTPCC_ODBC::CTPCC_ODBC (
        LPCSTR szServer,                                // name of SQL server
        LPCSTR szUser,                                          // user name
for login
        LPCSTR szPassword,                              // password for login
        LPCSTR szHost,                                          // not used
        LPCSTR szDatabase                               // name of database to
use
        )
{
        RETCODE            rc;

        // initialization
        m_hdbc = SQL_NULL_HDBC;
        m_hstmt = SQL_NULL_HSTMT;

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

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


        if ( SQLAllocHandle(SQL_HANDLE_DBC, henv, &m_hdbc) != SQL_SUCCESS )
                ThrowError(CODBCERR::eAllocHandle);

        if ( SQLSetConnectOption(m_hdbc, SQL_PACKET_SIZE, 4096) != SQL_SUCCESS )
                ThrowError(CODBCERR::eConnOption);

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

                sprintf( szConnectStr, "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
                        szServer, szUser, szPassword, szDatabase );

                rc = SQLDriverConnect(m_hdbc, NULL, (SQLCHAR*)szConnectStr,
sizeof(szConnectStr),
                        (SQLCHAR*)szOutStr, sizeof(szOutStr), &iOutStrLen,
SQL_DRIVER_NOPROMPT );

                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eConnect);
        }

        if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) != SQL_SUCCESS)
                ThrowError(CODBCERR::eAllocHandle);

        {
                char               buffer[128];

                // set some options affecting connection behavior
                strcpy(buffer, "set nocount on ");
                strcat(buffer, "set XACT_ABORT ON " );

                // for coyote
                strcat(buffer, "set ansi_warnings on " );
                strcat(buffer, "set ansi_nulls on " );

                rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eExecDirect);

                // verify that version of stored procs on server is correct
                char db_sp_version[10];
                strcpy(buffer, "{call tpcc_version}");
                rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
                if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                        ThrowError(CODBCERR::eExecDirect);
                if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR, &db_sp_version,
sizeof(db_sp_version), NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
                if ( SQLFetch(m_hstmt) == SQL_ERROR )
                        ThrowError(CODBCERR::eFetch);
                if (strcmp(db_sp_version,sVersion))
                        throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );

                SQLFreeHandle(SQL_HANDLE_STMT, m_hstmt);
        }

        // Bind parameters for each of the transactions
        InitNewOrderParams();
        InitPaymentParams();
        InitOrderStatusParams();
        InitDeliveryParams();
        InitStockLevelParams();
}

CTPCC_ODBC::~CTPCC_ODBC( void )
{
        // note: descriptors are automatically released when the connection is
dropped
```

```
	SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtNewOrder);
	SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtPayment);
	SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtDelivery);
	SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtOrderStatus);
	SQLFreeHandle(SQL_HANDLE_STMT, m_hstmtStockLevel);

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

void CTPCC_ODBC::ThrowError( CODBCERR::ACTION eAction )
{
	RETCODE		rc;
	SDWORD		lNativeError;
	char		szState[6];
	char		szMsg[SQL_MAX_MESSAGE_LENGTH];
	char		szTmp[6*SQL_MAX_MESSAGE_LENGTH];
	CODBCERR   *pODBCErr;				// not allocated until
needed (maybe never)

	pODBCErr = new CODBCERR();

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

	szTmp[0] = 0;
	while (TRUE)
	{
		rc = SQLError(henv, m_hdbc, m_hstmt, (BYTE *)&szState,
&lNativeError,
					(BYTE *)&szMsg, sizeof(szMsg),
NULL);
		if (rc == SQL_NO_DATA)
			break;

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

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

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

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

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

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

	SQLFreeStmt(m_hstmt, SQL_CLOSE);
	throw pODBCErr;
}

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

	m_hstmt = m_hstmtStockLevel;

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

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

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

	m_hstmt = m_hstmtStockLevel;

	while (TRUE)
	{
		try
		{
			rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_stocklevel(?,?,?)}", SQL_NTS);
			if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
				ThrowError(CODBCERR::eExecDirect);

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

			SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

			// hit deadlock; backoff for increasingly longer
period
```

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

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

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

        m_hstmt = m_hstmtNewOrder;

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

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

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

#ifndef new_order_strstr
        // set the bind offset pointer
        if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_ROW_BIND_OFFSET_PTR,
&m_BindOffset, SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);
        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_i_name, sizeof(m_txn.NewOrder.OL[0].ol_i_name), NULL) !=
SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.NewOrder.OL[0].ol_stock, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.NewOrder.OL[0].ol_brand_generic,
sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_i_price, 0, NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.NewOrder.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);
#else
        // prototype to eliminate patindex in server; shift work to client
        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_ol_i_name,
sizeof(m_ol_i_name), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT, &m_ol_stock, 0, NULL)
!= SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_i_data,
sizeof(m_i_data), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,   &m_s_data,
sizeof(m_s_data), NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_i_price, 0,
NULL) != SQL_SUCCESS
                        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_ol_amount, 0, NULL)
!= SQL_SUCCESS
                        )
                        ThrowError(CODBCERR::eBindCol);
#endif

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

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

void CTPCC_ODBC::NewOrder()
{
        int                                             i;
```

```
        RETCODE                                 rc;
        int                                             iTryCount = 0;

                  // 0         1         2

                  // 012345678901234567890123456789
        wchar_t                                 szSqlTemplate[] = L"{call
tpcc_neworder(?,?,?,?,?,?,"

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

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

                   L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}";

        m_hstmt = m_hstmtNewOrder;

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

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

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

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

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

                                      if ( SQLFetch(m_hstmt) == SQL_ERROR)
                                                ThrowError(CODBCERR::eFetch);
#else
                                      if ( SQLFetch(m_hstmt) == SQL_ERROR)
                                                ThrowError(CODBCERR::eFetch);
                                      strcpy( m_txn.NewOrder.OL[i].ol_i_name,
m_ol_i_name );
                                      if ( strstr(m_i_data, "ORIGINAL") != NULL &&
strstr(m_s_data, "ORIGINAL") != NULL )
                                      m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'B';
                                      else
                                      m_txn.NewOrder.OL[i].ol_brand_generic[0] = 'G';
                                      m_txn.NewOrder.OL[i].ol_brand_generic[1] =
0;
                                      m_txn.NewOrder.OL[i].ol_stock
                                      = m_ol_stock;
                                      m_txn.NewOrder.OL[i].ol_i_price
                                      = m_ol_i_price;
                                      m_txn.NewOrder.OL[i].ol_amount
                                      = m_ol_amount;
#endif
                                      // move to the next resultset
                                      if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
                                      ThrowError(CODBCERR::eMoreResults);
                                      m_txn.NewOrder.total_amount +=
m_txn.NewOrder.OL[i].ol_amount;
                            }

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

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

                            SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

```
                }
        }

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

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

        m_hstmt = m_hstmtPayment;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Payment.c_w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_DOUBLE,
SQL_NUMERIC, 6, 2, &m_txn.Payment.h_amount, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.d_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT,
SQL_C_UTINYINT, SQL_TINYINT, 0, 0, &m_txn.Payment.c_d_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Payment.c_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, sizeof(m_txn.Payment.c_last), 0, &m_txn.Payment.c_last,
sizeof(m_txn.Payment.c_last), NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,     &m_txn.Payment.c_id,
        0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_last,                sizeof(m_txn.Payment.c_last), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.h_date,                0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_1,   sizeof(m_txn.Payment.w_street_1), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_street_2,   sizeof(m_txn.Payment.w_street_2), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_city,                sizeof(m_txn.Payment.w_city), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_state,               sizeof(m_txn.Payment.w_state), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.w_zip,                 sizeof(m_txn.Payment.w_zip), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_1,   sizeof(m_txn.Payment.d_street_1), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_street_2,   sizeof(m_txn.Payment.d_street_2), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_city,                sizeof(m_txn.Payment.d_city), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_state,               sizeof(m_txn.Payment.d_state), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.d_zip,                 sizeof(m_txn.Payment.d_zip), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_first,               sizeof(m_txn.Payment.c_first), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_middle,     sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_1,   sizeof(m_txn.Payment.c_street_1), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_street_2,   sizeof(m_txn.Payment.c_street_2), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_city,                sizeof(m_txn.Payment.c_city), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_state,               sizeof(m_txn.Payment.c_state), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_zip,                 sizeof(m_txn.Payment.c_zip), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_phone,               sizeof(m_txn.Payment.c_phone), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.c_since,               0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_credit,     sizeof(m_txn.Payment.c_credit), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_credit_lim,0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_discount,   0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.Payment.c_balance,    0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.Payment.c_data,                sizeof(m_txn.Payment.c_data), NULL) !=
SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);
}

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

        m_hstmt = m_hstmtPayment;

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

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_payment(?,?,?,?,?,?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::eExecDirect);
```

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

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

                        if (m_txn.Payment.c_id == 0)
                                throw new CTPCC_ODBC_ERR(
CTPCC_ODBC_ERR::ERR_INVALID_CUST );
                        else
                                m_txn.Payment.exec_status_code = eOK;

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

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

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

void CTPCC_ODBC::InitOrderStatusParams()
{
        if (   SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtOrderStatus) !=
SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols1) != SQL_SUCCESS
                || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descOrderStatusCols2) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eAllocHandle);

        m_hstmt = m_hstmtOrderStatus;

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

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

        // configure block cursor
        if (   SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_BIND_TYPE,
(SQLPOINTER)sizeof(m_txn.OrderStatus.OL[0]), 0) != SQL_SUCCESS
                || SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROWS_FETCHED_PTR,
&m_RowsFetched, 0) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_supply_w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.OL[0].ol_i_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.OL[0].ol_quantity, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.OL[0].ol_amount, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.OL[0].ol_delivery_d, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);


        if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC,
m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )
                ThrowError(CODBCERR::eSetStmtAttr);

        i = 0;
        if (   SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0,
NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_last, sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_first, sizeof(m_txn.OrderStatus.c_first), NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,
&m_txn.OrderStatus.c_middle, sizeof(m_txn.OrderStatus.c_middle), NULL) !=
SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.OrderStatus.o_entry_d, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT,
&m_txn.OrderStatus.o_carrier_id, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,
&m_txn.OrderStatus.c_balance, 0, NULL) != SQL_SUCCESS
                || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG,
&m_txn.OrderStatus.o_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindCol);
}

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

        m_hstmt = m_hstmtOrderStatus;

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

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

        while (TRUE)
        {
                try
```

```
                {
                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)1, 0) != SQL_SUCCESS )
                                ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_orderstatus(?,?,?,?)}", SQL_NTS);
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                                ThrowError(CODBCERR::eExecDirect);

                        // configure block cursor
                        if ( SQLSetStmtAttrW(m_hstmt, SQL_ATTR_ROW_ARRAY_SIZE,
(SQLPOINTER)MAX_OL_ORDER_STATUS_ITEMS, 0) != SQL_SUCCESS )
                                ThrowError(CODBCERR::eSetStmtAttr);

                        rc = SQLFetchScroll( m_hstmt, SQL_FETCH_NEXT, 0 );
                        if ( ((rc == SQL_SUCCESS_WITH_INFO) && (m_RowsFetched
!= 0)) || (rc == SQL_ERROR) )
                                ThrowError(CODBCERR::eFetchScroll);

                        m_txn.OrderStatus.o_ol_cnt = (short)m_RowsFetched;

                        if (m_txn.OrderStatus.o_ol_cnt != 0)
                        {
                                if ( SQLSetStmtAttrW( m_hstmt,
SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols2, SQL_IS_POINTER ) != SQL_SUCCESS )

        ThrowError(CODBCERR::eSetStmtAttr);

                                if ( SQLMoreResults(m_hstmt) == SQL_ERROR )

        ThrowError(CODBCERR::eMoreResults);

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

                        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

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

        m_hstmt = m_hstmtDelivery;

        int i = 0;
        if (   SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
                || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
                )
                ThrowError(CODBCERR::eBindParam);

        for (i=0;i<10;i++)
        {
                if ( SQLBindCol(m_hstmt, (UWORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
                        ThrowError(CODBCERR::eBindCol);
        }
}

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

        m_hstmt = m_hstmtDelivery;

        while (TRUE)
        {
                try
                {
                        rc = SQLExecDirectW(m_hstmt, (SQLWCHAR*)L"{call
tpcc_delivery(?,?)}", SQL_NTS);
                        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
                                ThrowError(CODBCERR::eExecDirect);

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

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

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

```
                }
        }

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

tpcc_odbc.h

```
/*      FILE:           TPCC_ODBC.H
 *                              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 CODBCERR : public CBaseErr
{
        public:
                enum ACTION
                {
                        eNone,
                        eUnknown,
                        eAllocConn,                             // error from
SQLAllocConnect
                        eAllocHandle,               // error from
SQLAllocHandle
                        eConnOption,                // error from
SQLSetConnectOption
                        eConnect,                   // error from SQLConnect
                        eAllocStmt,                             // error from
SQLAllocStmt
                        eExecDirect,                // error from
SQLExecDirect
                        eBindParam,                             // error from
SQLBindParameter
                        eBindCol,                   // error from SQLBindCol
                        eFetch,                                 // error from
SQLFetch
                        eFetchScroll,               // error from
SQLFetchScroll
                        eMoreResults,               // error from
SQLMoreResults
                        ePrepare,                   // error from SQLPrepare
                        eExecute,                   // error from SQLExecute
                        eSetEnvAttr,                // error from
SQLSetEnvAttr
                        eSetStmtAttr                // error from
SQLSetStmtAttr
                };

                CODBCERR(void)
                {
                        m_eAction = eNone;
                        m_NativeError = 0;
                        m_bDeadLock = FALSE;
                        m_odbcerrstr = NULL;
                };

                ~CODBCERR()
                {
                        if (m_odbcerrstr != NULL)
                                delete [] m_odbcerrstr;
                };

                ACTION    m_eAction;
                int               m_NativeError;
                BOOL      m_bDeadLock;
                char   *m_odbcerrstr;

                int ErrorType() {return ERR_TYPE_ODBC;};
                int ErrorNum() {return m_NativeError;};
                char *ErrorText() {return m_odbcerrstr;};

};

class CTPCC_ODBC_ERR : public CBaseErr
{
        public:
                enum TPCC_ODBC_ERRS
                {
                        ERR_WRONG_SP_VERSION = 1,    // "Wrong version of
stored procs on database server"
                        ERR_INVALID_CUST,                       // "Invalid
Customer id,name."
                        ERR_NO_SUCH_ORDER,                      // "No orders
found for customer."
                        ERR_RETRIED_TRANS,                      // "Retries
before transaction succeeded."
                };

                CTPCC_ODBC_ERR( int iErr ) { m_errno = iErr; m_iTryCount = 0; };

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

                int               m_errno;
                int               m_iTryCount;

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

                char *ErrorText();
};

class DllDecl CTPCC_ODBC : public CTPCC_BASE
{
        private:
                // declare variables and private functions here...
```

```
                BOOL            m_bDeadlock;                // transaction
was selected as deadlock victim
                int                     m_MaxRetries;           //
retry count on deadlock

                SQLHENV         m_henv;                         //
ODBC environment handle
                SQLHDBC         m_hdbc;
                SQLHSTMT  m_hstmt;                  // the current hstmt

                SQLHSTMT  m_hstmtNewOrder;
                SQLHSTMT  m_hstmtPayment;
                SQLHSTMT  m_hstmtDelivery;
                SQLHSTMT  m_hstmtOrderStatus;
                SQLHSTMT  m_hstmtStockLevel;

                SQLHDESC  m_descNewOrderCols1;
                SQLHDESC  m_descNewOrderCols2;
                SQLHDESC  m_descOrderStatusCols1;
                SQLHDESC  m_descOrderStatusCols2;

                // new-order specific fields
                SQLUINTEGER     m_BindOffset;
                SQLUINTEGER     m_RowsFetched;
                int                     m_no_commit_flag;

#ifdef new_order_strstr
                // for new-order txn;
                // output params
                char            m_ol_i_name[I_NAME_LEN+1];
                double          m_ol_i_price;
                double          m_ol_amount;
                short           m_ol_stock;
                // used locally, but not returned to caller
                char            m_i_data[I_DATA_LEN];
                char            m_s_data[S_DATA_LEN];
#endif

                void ThrowError( CODBCERR::ACTION eAction );

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

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

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

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

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

};


// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new
        ( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase );

typedef CTPCC_ODBC* (TYPE_CTPCC_ODBC)(LPCSTR, LPCSTR, LPCSTR, LPCSTR, LPCSTR);
```

trans.h

```
/*        FILE:             TRANS.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 structure templates.
 *
 *  Change history:
 *                  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
        short                                   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
        short              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
        short                                   w_id;
        short                                   d_id;
        long                                    c_id;
        short                                   c_d_id;
        short                                   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;
        short                                   ol_supply_w_id;
        short                                   ol_quantity;
        double                                  ol_amount;
        TIMESTAMP_STRUCT   ol_delivery_d;
} OL_ORDER_STATUS_DATA;

typedef struct
```

```
{       // input params
        short               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
        short                       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
        short               w_id;                       //delivery warehouse
        short               o_carrier_id;       //carrier id
} DELIVERY_TRANSACTION;

typedef struct
{       // input params
        short                               w_id;
        short                               d_id;
        short                               threshold;

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

txn_base.h

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

#pragma once

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

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

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

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

txnlog.h

```
/*      FILE:               TXNLOG.H
 *                                  Microsoft TPC-C Kit Ver. 4.10.000
 *                                  not yet audited
 *
 *       PURPOSE:  Header file for txn log class
 *                                  Copyright Microsoft, 1999
 *             All Rights Reserved
 *
 */

#pragma once

    typedef struct _TXN_NEWORDER
          {
        BYTE        OL_Count;           //range 0 to 31
        BYTE        OL_Remote_Count;    //range 0 to 31
        WORD        c_id;
        int                 o_id;
          } TXN_NEWORDER;

    typedef struct _TXN_PAYMENT
          {
        BYTE        CustByName;
        BYTE        IsRemote;
```

```
        } TXN_PAYMENT;

    typedef struct _TXN_ORDERSTATUS
        {
                BYTE      CustByName;
        } TXN_ORDERSTATUS;

        typedef union _TXN_DETAILS
        {
                TXN_NEWORDER       NewOrder;
                TXN_PAYMENT                Payment;
                TXN_ORDERSTATUS    OrderStatus;
        } TXN_DETAILS;


        // Common header for all records in txn log.  The TxnType field is
        // a switch which identifies the particular variant.
        #define TXN_REC_TYPE_CONTROL          1        //
        #define TXN_REC_TYPE_TPCC             2        // replaces
TRANSACTION_TYPE_TPCC
        #define TXN_REC_TYPE_TPCC_DELIV_DEF   3

        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 CTXNLOG_ERRS
                {
                        ERR_BAD_FILE_FORMAT,                    // "File
format is invalid."
                        ERR_UNKNOWN_LOG_VERSION,     // "Log file version is
unknown."
                        ERR_BROKEN_LOG_FILE,                    // "Log file
is broken."
                        ERR_LOG_NOT_SORTED,                     // "Log file
is not sorted"
                        ERR_INVALID_TIME_SEQ,                   // "Internal
Error: Record Time Sequence invalid."
                };

                CTXNLOG_ERR(int iErr) : CBaseErr(iErr) {};

                int ErrorType() {return ERR_TYPE_TXNLOG;};

                char *ErrorText()
                {
                        static char *szMsgs[] = {
                                "File format is invalid.",
                                "Log file version is unknown.",
                                "Log file is broken.",
                                "Log file is not sorted",
                                "Internal Error: Record Time Sequence
invalid.",
                                ""
                        };

                        for(int i = 0; szMsgs[i][0]; i++)
                        {
                                if ( m_idMsg == i )
                                        break;
                        }

                        return(szMsgs[i][0] ? szMsgs[i] : ERR_UNKNOWN);
                };
};
```

Appendix B: Database Design

The TPC-C database was created with the following Transact-SQL scripts:

VerifyTpccLoad.sql

```
--  File:     VERIFYTPCCLOAD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Performs series of TPCC database checks to verify
--              that database load completed correctly

print     " "
select    convert(char(30), getdate(),9)
print     " "

use tpcc
go

--        ***********************************
--
--        Check rows per table from SYSINDEXES
--
--        ***********************************

print     'WAREHOUSE TABLE'

select    rows
from      sysindexes
where     id        = object_id("warehouse")
go

print     'DISTRICT TABLE = (10 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("district")
go

print     'ITEM TABLE = 100,000'

select    rows
from      sysindexes
where     id        =object_id("item")
go

print     'CUSTOMER TABLE = (30,000 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("customer")
go

print 'ORDERS TABLE = (30,000 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("orders")
go

print     'HISTORY TABLE = (30,000 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("history")
go

print     'STOCK TABLE = (100,000 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("stock")
go

print     'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'

select    rows
from      sysindexes
where     id        =object_id("order_line")
go

print     'NEW_ORDER TABLE = (9000 * No of warehouses)'

select    rows
from      sysindexes
where     id        =object_id("new_order")
go

--        *************
--
--        Check indices
--
--        *************

print '**************Index Check****************'

use tpcc
go

sp_helpindex       customer
go

sp_helpindex       stock
go

sp_helpindex       district
go

sp_helpindex       item
go

sp_helpindex       new_order
go
```

```
sp_helpindex       orders
go

sp_helpindex       order_line
go

sp_helpindex       warehouse
go
```

attachdb.sql

```
sp_attach_db @dbname = 'tpcc',
@filename1 = "C:\MSSQL70_tpcc_root.mdf",
@filename2 = "I:",
@filename3 = "J:",
@filename4 = "K:",
@filename5 = "M:",
@filename6 = "N:",
@filename7 = "O:",
@filename8 = "F:"
go
```

backup.sql

```
--  File:     BACKUP.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates backup of tpcc database


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

dump database tpcc to tpccback1, tpccback2, tpccback3 with init, stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

backupdev.sql

```
--  File:     BACKUPDEVB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates tpcc database Backup Devices


use master
go

--  create backup devices

exec sp_addumpdevice 'disk','tpccback1','x:\tpccback1.dmp'
go
exec sp_addumpdevice 'disk','tpccback2','y:\tpccback2.dmp'
go
exec sp_addumpdevice 'disk','tpccback3','z:\tpccback3.dmp'
go
```

createdb.sql

```
--  File:     CREATEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates tpcc database and backup files


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(),9))
go

--  create main database files

CREATE DATABASE tpcc
ON PRIMARY
(         NAME                = MSSQL70_tpcc_root,
          FILENAME  = "C:\MSSQL70_tpcc_root.mdf",
          SIZE                = 8MB,
          FILEGROWTH          =0),
FILEGROUP MSSQL70_misc_fg
(         NAME                = MSSQL70_misc1,
          FILENAME  = "I:",
          SIZE                = 18000MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL70_misc2,
          FILENAME  = "J:",
          SIZE                = 18000MB,
          FILEGROWTH          = 0),
(         NAME                = MSSQL70_misc3,
          FILENAME  = "K:",
          SIZE                = 18000MB,
          FILEGROWTH          = 0),
FILEGROUP MSSQL70_cs_fg
(         NAME                = MSSQL70_cs1,
          FILENAME  = "M:",
```

```
        SIZE                = 35000MB,
        FILEGROWTH          = 0),
(       NAME                = MSSQL70_cs2,
        FILENAME  = "N:",
        SIZE                = 35000MB,
        FILEGROWTH          = 0),
(       NAME                = MSSQL70_cs3,
        FILENAME  = "O:",
        SIZE                = 35000MB,
        FILEGROWTH          = 0)
LOG ON
(       NAME                =MSSQL70_tpcc_log,
        FILENAME  ="F:",
        SIZE                =52000MB,
        FILEGROWTH          =0)
go

-- Store ending time
update    tpcc_timer
set       end_date  = (select convert(char(30), getdate(),9))
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
```

dbopt1.sql

```
--  File:     DBOPT1.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Sets database options for data load


use master
go

exec sp_dboption tpcc,'select into/bulkcopy',true
exec sp_dboption tpcc,'trunc. log on chkpt.',true
exec sp_dboption tpcc,'torn page detection',false
go

use tpcc
go

checkpoint
go
```

dbopt2.sql

```
--  File:     DBOPT2.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Resets database options after data load

sp_dboption tpcc,'select into/bulkcopy',FALSE
GO

sp_dboption tpcc,'trunc. log on chkpt.',FALSE
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE   @msg               varchar(50)

IF (SELECT (SUBSTRING((SELECT @@version),1,25))) = 'Microsoft SQL Server 2000'
        BEGIN
                --                                                  --
                --        OPTIONS FOR SQL SERVER 8.0                --
                -- Set option values for user-defined indexes --
                --                                                  --

                SET       @msg      = ' '
                PRINT     @msg                                       --
                SET       @msg      = 'Setting SQL Server 8.0 indexoptions'
                PRINT     @msg
                SET       @msg      = ' '
                PRINT     @msg                                        --

                EXEC sp_indexoption'customer',        'DisAllowPageLocks',
        TRUE
                EXEC sp_indexoption'district',        'DisAllowPageLocks',
        TRUE
                EXEC sp_indexoption'warehouse',       'DisAllowPageLocks',
        TRUE
                EXEC sp_indexoption'stock',  'DisAllowPageLocks',         TRUE
                EXEC sp_indexoption'order_line',      'DisAllowRowLocks',TRUE
                EXEC sp_indexoption'orders', 'DisAllowRowLocks',TRUE
                EXEC sp_indexoption'new_order',       'DisAllowRowLocks',TRUE
                EXEC sp_indexoption'item',            'DisAllowRowLocks',TRUE
                EXEC sp_indexoption'item',            'DisAllowPageLocks',
        TRUE
        END
ELSE
        BEGIN
                --                                                  --
                --        OPTIONS FOR SQL SERVER 7.0                --
                -- Set option values for user-defined indexes --
                --

                SET       @msg      = ' '
                PRINT     @msg                                       --
                SET       @msg      = 'Setting SQL Server 7.0 indexoptions'
                PRINT     @msg
```

```
                SET     @msg    = ' '
                PRINT   @msg                                    --

                EXEC sp_indexoption'customer',        'AllowPageLocks',
        FALSE
                EXEC sp_indexoption'district',        'AllowPageLocks',
        FALSE
                EXEC sp_indexoption'warehouse',       'AllowPageLocks',
        FALSE
                EXEC sp_indexoption'stock',  'AllowPageLocks',  FALSE
                EXEC sp_indexoption'order_line',      'AllowRowLocks',
        FALSE
                EXEC sp_indexoption'orders', 'AllowRowLocks',   FALSE
                EXEC sp_indexoption'new_order',       'AllowRowLocks',
        FALSE
                EXEC sp_indexoption'item',            'AllowRowLocks',
        FALSE
                EXEC sp_indexoption'item',            'AllowPageLocks',
        FALSE

        END
GO

Print ' '
Print '******************'
Print 'Pre-specified Locking Hierarchy:'
Print '     Lockflag = 0 ==> No pre-specified hierarchy'
Print '   Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '   Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '   Lockflag = 3 ==> Lock at Table-level'
Print ' '

SELECT   name,lockflags
FROM     sysindexes
WHERE    object_id('warehouse')       = id OR
         object_id('district')        = id OR
         object_id('customer')        = id OR
         object_id('stock') = id OR
         object_id('orders')= id OR
         object_id('order_line')      = id OR
         object_id('history')         = id OR
         object_id('new_order')       = id OR
         object_id('item')  = id
ORDER    BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

EXEC sp_dboption tpcc,       'auto update statistics',    FALSE
EXEC sp_dboption tpcc,       'auto create statistics',    FALSE
GO

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
```

delivery.sql

```
--  File:     DELIVERY.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates delivery transaction stored procedure
--
--        Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_delivery    @w_id           smallint,

                                         @o_carrier_id   smallint
as

declare @d_id      tinyint,
        @o_id      int,
        @c_id      int,
        @total     numeric(12,2),
        @oid1      int,
        @oid2      int,
        @oid3      int,
        @oid4      int,
        @oid5      int,
        @oid6      int,
        @oid7      int,
        @oid8      int,
        @oid9      int,
        @oid10     int

select @d_id = 0

begin tran d

         while (@d_id < 10)
         begin

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

                  select    top 1
                            @o_id     = no_o_id
                  from      new_order (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 tran d

-- return delivery data to client

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

go
```

getargs.c

```
//        File:              GETARGS.C
//                                   Microsoft TPC-C Kit Ver. 4.21
//                                   Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//        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->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 'h':     /* Fall throught */
                case 'H':
                                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 '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("-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);
}
```

idxcuscl.sql

```
--  File:     IDXCUSCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)
go
```

idxcusnc.sql

```
--  File:     IDXCUSNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates non-clustered index on customer table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxdiscl.sql

```
--  File:     IDXDISCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on district table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxitmcl.sql

```
--  File:     IDXITMCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on item table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxnodcl.sql

```
--  File:     IDXNODCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on new_order table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxodlcl.sql

```
--  File:     IDXODLCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on order_line table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordcl.sql

```
--  File:     IDXORDCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

if exists ( select name from sysindexes where  name = 'orders_c1' )
         drop index orders.orders_c1

create unique clustered index orders_c1 on orders(o_w_id, o_d_id, o_id)
         on MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxordnc.sql

```
--  File:     IDXORDNC.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates non-clustered index on orders table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxstkcl.sql

```
--  File:     IDXSTKCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on stock table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_cs_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

idxwarcl.sql

```
--  File:     IDXWARCL.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates clustered index on warehouse table


use tpcc
go

declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

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 MSSQL70_misc_fg

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

neword.sql

```
--  File:     NEWORD.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates new order transaction stored procedure
--
--        Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_neworder
                                                @w_id           smallint,
                                                @d_id           tinyint,
                                                @c_id           int,
                                                @o_ol_cnt       tinyint,
                                                @o_all_local    tinyint,
                                                @i_id1  int = 0, @s_w_id1
smallint = 0, @ol_qty1  smallint = 0,
                                                @i_id2  int = 0, @s_w_id2
smallint = 0, @ol_qty2  smallint = 0,
                                                @i_id3  int = 0, @s_w_id3
smallint = 0, @ol_qty3  smallint = 0,
                                                @i_id4  int = 0, @s_w_id4
smallint = 0, @ol_qty4  smallint = 0,
```

```
smallint = 0, @ol_qty5  smallint = 0,
smallint = 0, @ol_qty6  smallint = 0,
smallint = 0, @ol_qty7  smallint = 0,
smallint = 0, @ol_qty8  smallint = 0,
smallint = 0, @ol_qty9  smallint = 0,
smallint = 0, @ol_qty10 smallint = 0,
smallint = 0, @ol_qty11 smallint = 0,
smallint = 0, @ol_qty12 smallint = 0,
smallint = 0, @ol_qty13 smallint = 0,
smallint = 0, @ol_qty14 smallint = 0,
smallint = 0, @ol_qty15 smallint = 0


as
declare   @w_tax          numeric(4,4),
        @d_tax          numeric(4,4),
        @c_last         char(16),
        @c_credit       char(2),
        @c_discount     numeric(4,4),
        @i_price        numeric(5,2),
        @i_name         char(24),
        @i_data         char(50),
        @o_entry_d      datetime,
        @remote_flag    int,
        @s_quantity     smallint,
        @s_data         char(50),
        @s_dist         char(24),
          @li_no          int,
          @o_id               int,
          @commit_flag    tinyint,
        @li_id          int,
        @li_s_w_id      smallint,
        @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
```

```
@i_id5  int = 0, @s_w_id5
@i_id6  int = 0, @s_w_id6
@i_id7  int = 0, @s_w_id7
@i_id8  int = 0, @s_w_id8
@i_id9  int = 0, @s_w_id9
@i_id10 int = 0, @s_w_id10
@i_id11 int = 0, @s_w_id11
@i_id12 int = 0, @s_w_id12
@i_id13 int = 0, @s_w_id13
@i_id14 int = 0, @s_w_id14
@i_id15 int = 0, @s_w_id15
```

```
--  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 (tablock 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,
                                                            @li_s_w_id,
                                                            "dec 31, 1899",
                                                            @li_qty,
                                                            @i_price *
@li_qty,
                                                            @s_dist)

--  send line-item data to client

                              select    @i_name,
                                        @s_quantity,
                                        b_g = case when (
(patindex("%ORIGINAL%",@i_data) > 0) and
(patindex("%ORIGINAL%",@s_data) > 0) )
                                                             then "B" else "G" end,
                                        @i_price,
                                        @i_price * @li_qty

                    end
                    else
                    begin

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

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

                    end
          end

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

          select    @c_last     = c_last,
                    @c_discount = c_discount,
                    @c_credit   = c_credit,
                    @c_id_local = c_id
          from      customer (repeatableread)
          where     c_id                = @c_id and
                    c_w_id              = @w_id and
                    c_d_id              = @d_id

--  insert fresh row into orders table

          insert into orders values (  @o_id,
                                                        @d_id,
                                                        @w_id,
                                                        @c_id_local,
                                                        @o_entry_d,
                                                        0,
                                                        @o_ol_cnt,
                                                        @o_all_local)

--  insert corresponding row into new-order table

          insert into new_order values (        @o_id,
                                                        @d_id,
                                                        @w_id)

--  select warehouse tax

          select    @w_tax    = w_tax
          from      warehouse (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
```

ordstat.sql

```
--  File:     ORDSTAT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates order status transaction stored procedure
--
--        Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_orderstatus @w_id     smallint,
                                       @d_id     tinyint,

                                       @c_id     int,
                                       @c_last   char(16) = ""


as

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

begin tran o

if (@c_id = 0)
         begin

--  get customer id and info using last name

                  select    @cnt      = (count(*)+1)/2
                  from      customer (repeatableread)
                  where     c_last    = @c_last and
                            c_w_id    = @w_id and
                            c_d_id    = @d_id

                  set       rowcount @cnt

                  select    @c_id              = c_id,
                            @c_balance         = c_balance,
                               @c_first        = c_first,
                               @c_last         = c_last,
                               @c_middle       = c_middle
                  from      customer (repeatableread)
                  where     c_last             = @c_last and
                            c_w_id             = @w_id and
                            c_d_id             = @d_id
                  order     by c_w_id, c_d_id, c_last, c_first

                  set       rowcount 0
         end

         else

         begin

--  get customer info if by id

                  select    @c_balance         = c_balance,
                            @c_first  = c_first,
                            @c_middle = c_middle,
                            @c_last            = c_last
                  from      customer (repeatableread)
                  where     c_id               = @c_id and
                            c_d_id             = @d_id and
                            c_w_id             = @w_id

                  select    @cnt      = @@rowcount

         end

--  if no such customer

         if (@cnt = 0)
         begin
                  raiserror("Customer not found",18,1)
                  goto custnotfound
         end

--  get order info

         select    @o_id              = o_id,
                   @o_entry_d         = o_entry_d,
                   @o_carrier_id      = o_carrier_id
         from      orders (serializable)
         where     o_c_id             = @c_id and
                   o_d_id             = @d_id and
                   o_w_id             = @w_id
         order     by o_id asc

--  select order lines for the current order

         select    ol_supply_w_id,
                   ol_i_id,
                   ol_quantity,
                   ol_amount,
                   ol_delivery_d
         from      order_line (repeatableread)
         where     ol_o_id = @o_id and
                   ol_d_id = @d_id and
                   ol_w_id = @w_id

custnotfound:
```

```
commit tran o

--  return data to client

select    @c_id,
          @c_last,
          @c_first,
          @c_middle,
          @o_entry_d,
          @o_carrier_id,
          @c_balance,
          @o_id

go
```

payment.sql

```
--  File:     PAYMENT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates payment transaction stored procedure
--
--        Interface Level: 4.10.000

use tpcc
go

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

create proc tpcc_payment     @w_id           smallint,
                                    @c_w_id         smallint,
                                    @h_amount       numeric(6,2),
                                    @d_id           tinyint,
                                    @c_d_id         tinyint,
                                    @c_id           int,
                                    @c_last         char(16) = ""



as
declare  @w_street_1     char(20),
          @w_street_2     char(20),
          @w_city         char(20),
          @w_state        char(2),
          @w_zip          char(9),
          @w_name         char(10),
          @d_street_1     char(20),
          @d_street_2     char(20),
          @d_city         char(20),
          @d_state        char(2),
          @d_zip          char(9),
          @d_name         char(10),
          @c_first        char(16),
          @c_middle       char(2),
          @c_street_1     char(20),
          @c_street_2     char(20),
          @c_city         char(20),
          @c_state        char(2),
          @c_zip          char(9),
          @c_phone        char(16),
          @c_since        datetime,
          @c_credit       char(2),
          @c_credit_lim   numeric(12,2),
          @c_balance      numeric(12,2),
          @c_discount     numeric(4,4),
          @data           char(500),
          @c_data         char(500),
          @datetime       datetime,
          @w_ytd          numeric(12,2),
          @d_ytd          numeric(12,2),
          @cnt            smallint,
          @val            smallint,
          @screen_data         char(200),
            @d_id_local        tinyint,
            @w_id_local        smallint,
            @c_id_local        int

select @screen_data = ""

begin tran p

--  get payment date

          select    @datetime = getdate()

          if (@c_id = 0)
          begin

--  get customer id and info using last name

                    select    @cnt      = count(*)
                    from      customer (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 (repeatableread)
                    where     c_last    = @c_last and
                              c_w_id    = @c_w_id and
                              c_d_id    = @c_d_id
                    order     by c_last, c_first

                    set       rowcount 0
          end

--  get customer info and update balances

          update    customer
          set       @c_balance          = c_balance        = c_balance - @h_amount,
                    c_payment_cnt       = c_payment_cnt + 1,
                    c_ytd_payment       = c_ytd_payment + @h_amount,
                    @c_first  = c_first,
                    @c_middle = c_middle,
                    @c_last             = c_last,
                    @c_street_1         = c_street_1,
                    @c_street_2         = c_street_2,
                    @c_city             = c_city,
                    @c_state  = c_state,
                    @c_zip              = c_zip,
```

```
                @c_phone  = c_phone,
                @c_credit = c_credit,
                @c_credit_lim      = c_credit_lim,
                @c_discount        = c_discount,
                @c_since  = c_since,
                @data              = c_data,
                @c_id_local        = c_id
        where   c_id               = @c_id and
                c_w_id             = @c_w_id and
                c_d_id             = @c_d_id

--  if customer has bad credit get some more info

        if (@c_credit = "BC")
        begin

--      compute new info

                select @c_data     = convert(char(5),@c_id) +
                                     convert(char(4),@c_d_id) +
                                     convert(char(5),@c_w_id) +
                                     convert(char(4),@d_id) +
                                     convert(char(5),@w_id) +
                                     convert(char(19),@h_amount) +
                                     substring(@data, 1, 458)

--  update customer info

                update  customer
                set     c_data  = @c_data
                where   c_id    = @c_id and
                        c_w_id  = @c_w_id and
                        c_d_id  = @c_d_id

                select  @screen_data = substring (@c_data,1,200)
        end

--  get district data and update year-to-date

        update  district
        set     d_ytd              = d_ytd + @h_amount,
                @d_street_1        = d_street_1,
                @d_street_2        = d_street_2,
                @d_city            = d_city,
                @d_state  = d_state,
                @d_zip             = d_zip,
                @d_name            = d_name,
                @d_id_local        = d_id
        where   d_w_id             = @w_id and
                d_id               = @d_id

--  get warehouse data and update year-to-date

        update  warehouse
        set     w_ytd              = w_ytd + @h_amount,
                @w_street_1        = w_street_1,
                @w_street_2        = w_street_2,
                @w_city            = w_city,
                @w_state  = w_state,
                @w_zip             = w_zip,
                @w_name            = w_name,
                @w_id_local        = w_id
        where   w_id               = @w_id

--  create history record

        insert into history values ( @c_id_local,
                                                @c_d_id,
                                                @c_w_id,
                                                @d_id_local,
                                                @w_id_local,
                                                @datetime,
                                                @h_amount,
                                                @w_name + "    " + @d_name)
commit tran p

--  return data to client

select  @c_id,
        @c_last,
        @datetime,
        @w_street_1,
        @w_street_2,
        @w_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
```

random.c

```
//      File:           RANDOM.C
//                                  Microsoft TPC-C Kit Ver. 4.21
//                                  Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//      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;
}
```

removedb.sql

```
--  File:     REMOVEDB.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Removes tpcc database and backup files


use master
go

--  remove any existing database and backup files

exec sp_dbremove tpcc, dropdev
go

exec sp_dropdevice 'tpccback1'
exec sp_dropdevice 'tpccback2'
exec sp_dropdevice 'tpccback3'
go
```

restore.sql

```
--  File:     RESTORE.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Loads database backup from backup files


declare @startdate datetime
declare @enddate datetime
select @startdate = getdate()
select "Start date:", convert(varchar(30),@startdate,9)

restore database tpcc from tpccback1, tpccback2, tpccback3 with replace, stats = 1

select @enddate = getdate()
select "End date: ", convert(varchar(30),@enddate,9)
select "Elapsed time (in seconds): ", datediff(second, @startdate, @enddate)

go
```

runcfg80.sql

```
/*  TPC-C Benchmark Kit                                                        */
/*                                                                             */
/*  RUNCFG80.SQL                                                               */
/*                                                                             */
/*  This script file is used to set runtime server configuration parameters    */
/*                                                                             */


exec sp_configure "show advanced option", 1
```

```
go

reconfigure with override
go

/* change this value to approximately the number of connected users */
exec sp_configure "max worker threads",255

/* increase priority of user threads */
exec sp_configure "priority boost",1

/* disable automatic checkpointing */
exec sp_configure "recovery interval",32767

/* change to a mask appropriate for the number of processors on the server */
exec sp_configure "affinity mask",0xf

/* enable fibers */
exec sp_configure "lightweight pooling",1

go

reconfigure with override
go
```

sqlshutdown.sql

```
use tpcc
go
shutdown
go
```

stocklev.sql

```
--  File:     STOCKLEV.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates stock level transaction stored procedure
--
--        Interface Level: 4.10.000

use tpcc
go

if exists (select name from sysobjects where name = "tpcc_stocklevel" )
         drop procedure tpcc_stocklevel
go

create proc tpcc_stocklevel  @w_id              smallint,
                                    @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

go
```

strings.c

```
//        File:               STRINGS.C
//                                    Microsoft TPC-C Kit Ver. 4.21
//                                    Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//        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++)
        {
                cc = chArray[RandomNumber(0, chArrayMax)];
       str[i] = cc;
    }
        if ( len < z )
                memset(str+len, ' ', z - len);
        str[len] = 0;

    return len;
}


//============================================================================
//
// Function name: MakeOriginalAlphaString
//
//============================================================================

int MakeOriginalAlphaString(int  x,
                                                        int  y,
                                                             int  z,
                                                        char *str,
                                                    int  percent)
{
    int         len;
    int         val;
    int         start;

#ifdef DEBUG
    printf("[%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 + y) <= 8)
    {
                printf("MakeOriginalAlphaString: string length must be >= 8\n");
                exit(-1);
    }

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

    val = RandomNumber(1,100);
    if (val <= percent)
    {
                start = RandomNumber(0, len - 8);
                strncpy(str + start, "ORIGINAL", 8);
    }


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

    return strlen(str);
}


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

         //MakeNumberString is always called MakeZipNumberString(16, 16, 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;
}
```

tables.sql

```
--  File:     TABLES.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Creates TPC-C tables


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                                    smallint,
        w_name                                  char(10),
        w_street_1                              char(20),
        w_street_2                              char(20),
        w_city                                  char(20),
        w_state                                 char(2),
        w_zip                                   char(9),
        w_tax                                   numeric(4,4),
        w_ytd                                   numeric(12,2)
) on MSSQL70_misc_fg
go

create table district
(
        d_id                                    tinyint,
        d_w_id                                  smallint,
        d_name                                  char(10),
        d_street_1                              char(20),
        d_street_2                              char(20),
        d_city                                  char(20),
        d_state                                 char(2),
        d_zip                                   char(9),
        d_tax                                   numeric(4,4),
        d_ytd                                   numeric(12,2),
        d_next_o_id                             int
) on MSSQL70_misc_fg
go

create table customer
(
        c_id                                    int,
        c_d_id                                  tinyint,
        c_w_id                                  smallint,
        c_first                                 char(16),
        c_middle                        char(2),
        c_last                                  char(16),
        c_street_1                              char(20),
        c_street_2                              char(20),
        c_city                                  char(20),
        c_state                                 char(2),
        c_zip                                   char(9),
        c_phone                                 char(16),
        c_since                                 datetime,
        c_credit                        char(2),
        c_credit_lim                    numeric(12,2),
        c_discount                              numeric(4,4),
        c_balance                       numeric(12,2),
        c_ytd_payment                   numeric(12,2),
        c_payment_cnt                   smallint,
        c_delivery_cnt                  smallint,
        c_data                                  char(500)
) on MSSQL70_cs_fg
go

create table history
(
        h_c_id                                  int,
        h_c_d_id                        tinyint,
        h_c_w_id                        smallint,
        h_d_id                                  tinyint,
        h_w_id                                  smallint,
        h_date                                  datetime,
        h_amount                        numeric(6,2),
        h_data                                  char(24)
) on MSSQL70_misc_fg
go
```

```
create table new_order
(
        no_o_id                         int,
        no_d_id                         tinyint,
        no_w_id                         smallint
) on MSSQL70_misc_fg
go

create table orders
(
        o_id                            int,
        o_d_id                          tinyint,
        o_w_id                          smallint,
        o_c_id                          int,
        o_entry_d               datetime,
        o_carrier_id            tinyint,
        o_ol_cnt                tinyint,
        o_all_local                     tinyint
) on MSSQL70_misc_fg
go

create table order_line
(
        ol_o_id                         int,
        ol_d_id                         tinyint,
        ol_w_id                         smallint,
        ol_number               tinyint,
        ol_i_id                         int,
        ol_supply_w_id          smallint,
        ol_delivery_d           datetime,
        ol_quantity                     smallint,
        ol_amount               numeric(6,2),
        ol_dist_info            char(24)
) on MSSQL70_misc_fg
go

create table item
(
        i_id                            int,
        i_im_id                         int,
        i_name                          char(24),
        i_price                         numeric(5,2),
        i_data                          char(50)
) on MSSQL70_misc_fg
go

create table stock
(
        s_i_id                          int,
        s_w_id                          smallint,
        s_quantity                      smallint,
        s_dist_01               char(24),
        s_dist_02               char(24),
        s_dist_03               char(24),
        s_dist_04               char(24),
        s_dist_05               char(24),
        s_dist_06               char(24),
        s_dist_07               char(24),
        s_dist_08               char(24),
        s_dist_09               char(24),
        s_dist_10               char(24),
        s_ytd                           int,
        s_order_cnt                     smallint,
        s_remote_cnt                    smallint,
        s_data                                  char(50)
) on MSSQL70_cs_fg
go
```

time.c

```
//      File:           TIME.C
//                                      Microsoft TPC-C Kit Ver. 4.21
//                                      Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//      Purpose:  Source file for time functions


// Includes
#include "tpcc.h"

// Globals
static long start_sec;


//===============================================================================
//
// Function name: TimeNow
//
//===============================================================================

long TimeNow()
{
   long            time_now;
   struct _timeb el_time;

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

   _ftime(&el_time);

   time_now = ((el_time.time - start_sec) * 1000) + el_time.millitm;

   return time_now;
}
```

tpcc.h

```
//      File:           TPCC.H
//                                      Microsoft TPC-C Kit Ver. 4.21
//                                      Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//      Purpose:  Header file for TPC-C database loader


// Build number of TPC Benchmark Kit
#define TPCKIT_VER "4.21"

// 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>

// 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                   "logs\\load.out"
#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                               *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       MakeOriginalAlphaString();
```

```
int       MakeNumberString();
int       MakeZipNumberString();
void      InitString();
void      InitAddress();
void      PaddString();
```

tpccldr.c

```
//        File:             TPCCLDR.C
//                                  Microsoft TPC-C Kit Ver. 4.21
//                                  Copyright Microsoft, 1996, 1997, 1998, 1999,
2000
//        Purpose:  Source file for TPC-C database loader


// Includes
#include "tpcc.h"
#include "search.h"

// Defines
#define MAXITEMS                100000
#define MAXITEMS_SCALE_DOWN             100
#define CUSTOMERS_PER_DISTRICT  3000
#define CUSTOMERS_SCALE_DOWN 30
#define DISTRICT_PER_WAREHOUSE  10
#define ORDERS_PER_DISTRICT     3000
#define ORDERS_SCALE_DOWN       30
#define MAX_CUSTOMER_THREADS    2
#define MAX_ORDER_THREADS       3
#define MAX_MAIN_THREADS        4

// Functions declarations

void HandleErrorDBC (SQLHDBC  hdbc1);

void CheckSQL();
void CheckDataBase();

long NURand();
void LoadItem();
void LoadWarehouse();

void Stock();
void District();

void LoadCustomer();
void CustomerBufInit();
void CustomerBufLoad();
void LoadCustomerTable();
void LoadHistoryTable();

void LoadOrders();
void OrdersBufInit();
void OrdersBufLoad();
void LoadOrdersTable();
void LoadNewOrderTable();
void LoadOrderLineTable();
void GetPermutation();
void CheckForCommit();
void OpenConnections();
void BuildIndex();
void FormatDate ();
```

```
// Shared memory structures

typedef struct
{
    long                ol;
    long                ol_i_id;
    short               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;
    short               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;
    short                             c_w_id;
         char                                 c_first[FIRST_NAME_LEN+1];
    char                              c_middle[MIDDLE_NAME_LEN+1];
    char                              c_last[LAST_NAME_LEN+1];
    char                              c_street_1[ADDRESS_LEN+1];
    char                              c_street_2[ADDRESS_LEN+1];
    char                              c_city[ADDRESS_LEN+1];
    char                              c_state[STATE_LEN+1];
    char                              c_zip[ZIP_LEN+1];
    char                              c_phone[PHONE_LEN+1];
    char                              c_credit[CREDIT_LEN+1];
    double                                   c_credit_lim;
    double                                   c_discount;
//  fix to avoid ODBC float to numeric conversion problem.
//  double                                   c_balance;
         char                                c_balance[6];

    double                                   c_ytd_payment;
         short                               c_payment_cnt;
         short                               c_delivery_cnt;
         char                                c_data[C_DATA_LEN+1];
         double                              h_amount;
    char                              h_data[H_DATA_LEN+1];
} CUSTOMER_STRUCT;

typedef struct
{
         char                                c_last[LAST_NAME_LEN+1];
         char                                c_first[FIRST_NAME_LEN+1];
    long                              c_id;
} CUSTOMER_SORT_STRUCT;
```

```
typedef struct
{
    long               time_start;
} LOADER_TIME_STRUCT;


// Global variables

char      szLastError[300];

HENV      henv;

HDBC      v_hdbc;                                          // for SQL
Server version verification
HDBC      i_hdbc1;                                  // for ITEM table
HDBC      w_hdbc1;                                  // for WAREHOUSE,
DISTRICT, STOCK
HDBC      c_hdbc1;                                  // for CUSTOMER
HDBC      c_hdbc2;                                  // for HISTORY
HDBC      o_hdbc1;                                  // for ORDERS
HDBC      o_hdbc2;                                  // for NEW-ORDER

HDBC      o_hdbc3;                                  // for ORDER-LINE

HSTMT     v_hstmt;                                  // for SQL Server
version verification
HSTMT     i_hstmt1;
HSTMT     w_hstmt1;
HSTMT     c_hstmt1, c_hstmt2;
HSTMT     o_hstmt1, o_hstmt2, o_hstmt3;

ORDERS_STRUCT   orders_buf[ORDERS_PER_DISTRICT];
CUSTOMER_STRUCT customer_buf[CUSTOMERS_PER_DISTRICT];
long            orders_rows_loaded;
long            new_order_rows_loaded;
long            order_line_rows_loaded;
long            history_rows_loaded;
long            customer_rows_loaded;
long            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);

      // verify database and tables exist before attempting to load

      CheckDataBase();

      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);

        printf("%s",buffer);
        fprintf(fLoader,"%s",buffer);

        main_time_start = (TimeNow() / MILLI);

        // start parallel load threads


        if (aptr->tables_all || aptr->table_item)
        {
                fprintf(fLoader, "\nStarting loader threads for: item\n");

                hThread[0] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadItem,
NULL,
                                                                        0,
&dwThreadID[0]);

                if (hThread[0] == NULL)
                {
                        printf("Error, failed in creating creating thread =
0.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_warehouse)
        {
                fprintf(fLoader, "Starting loader threads for: warehouse\n");

                hThread[1] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadWarehouse,
NULL,
                                                                        0,
&dwThreadID[1]);

                if (hThread[1] == NULL)
                {
                        printf("Error, failed in creating creating thread =
1.\n");
                        exit(-1);
                }

        }

        if (aptr->tables_all || aptr->table_customer)
        {
                fprintf(fLoader, "Starting loader threads for: customer\n");

                hThread[2] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadCustomer,
NULL,
                                                                        0,
&dwThreadID[2]);

                if (hThread[2] == NULL)
                {
                        printf("Error, failed in creating creating main thread
= 2.\n");
                        exit(-1);
                }
        }

        if (aptr->tables_all || aptr->table_orders)
        {
                fprintf(fLoader, "Starting loader threads for: orders\n");

                hThread[3] = CreateThread(NULL,
                                                                        0,
(LPTHREAD_START_ROUTINE) LoadOrders,
NULL,
                                                                        0,
&dwThreadID[3]);

                if (hThread[3] == NULL)
                {
                        printf("Error, failed in creating creating main thread
= 3.\n");
                        exit(-1);
                }
        }

        // Wait for threads to finish...
        for (i=0; i<MAX_MAIN_THREADS; i++)
        {
                if (hThread[i] != NULL)
                {
                        WaitForSingleObject( hThread[i], INFINITE );
                        CloseHandle(hThread[i]);
                        hThread[i] = NULL;
                }
        }

        main_time_end = (TimeNow() / MILLI);

    sprintf(buffer,"\nTPC-C load completed successfully in %ld minutes.\n",
                        (main_time_end - main_time_start)/60);

        printf("%s",buffer);
```

```
        fprintf(fLoader, "%s", buffer);

        fclose(fLoader);

        SQLFreeEnv(henv);

    exit(0);

        return 0;
}


//======================================================================
//
// Function name: LoadItem
//
//======================================================================

void LoadItem()
{
    long            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];

        // Seed with unique number
        seed(1);

        printf("Loading item table...\n");

        // if build index before load
        if ((aptr->build_index == 1) && (aptr->index_order == 1))
                BuildIndex("idxitmcl");

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

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

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

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

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

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

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

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

        time_start = (TimeNow() / MILLI);

        item_rows_loaded = 0;

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

                MakeAlphaString(14, 24, I_NAME_LEN, i_name);

                i_price = ((float) RandomNumber(100L, 10000L))/100.0;

                MakeOriginalAlphaString(26, 50, I_DATA_LEN, i_data, 10);

                rc = bcp_sendrow(i_hdbc1);
                if (rc != SUCCEED)
                        HandleErrorDBC(i_hdbc1);

                item_rows_loaded++;
                CheckForCommit(i_hdbc1, i_hstmt1, item_rows_loaded, "item",
&time_start);
        }

        rcint = bcp_done(i_hdbc1);
        if (rcint < 0)
                HandleErrorDBC(i_hdbc1);

        printf("Finished loading item table.\n");

        SQLFreeStmt(i_hstmt1, SQL_DROP);
        SQLDisconnect(i_hdbc1);
        SQLFreeConnect(i_hdbc1);

        // if build index after load
        if ((aptr->build_index == 1) && (aptr->index_order == 0))
                BuildIndex("idxitmcl");

}


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

```
//
//============================================================================

void LoadWarehouse()
{
    short 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];


          // 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");

          rc = bcp_init(w_hdbc1, name, NULL, "logs\\whouse.err", 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);
          }

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

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

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

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

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

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

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

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

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

          time_start = (TimeNow() / MILLI);

          warehouse_rows_loaded = 0;

          for (w_id = (short)aptr->starting_warehouse; w_id <= aptr->num_warehouses;
w_id++)
          {
                    MakeAlphaString(6,10, W_NAME_LEN, w_name);

                    MakeAddress(w_street_1, w_street_2, w_city, w_state, w_zip);

                    w_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                    w_ytd = 300000.00;

                    rc = bcp_sendrow(w_hdbc1);
                    if (rc != 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()
{
    short d_id;
    short 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;
          int                w_id;
          RETCODE  rc;
          DBINT    rcint;
          char     bcphint[128];

          // 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");

          rc = bcp_init(w_hdbc1, name, NULL, "logs\\district.err", 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);
          }

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

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 2);
          if (rc != SUCCEED)
                    HandleErrorDBC(w_hdbc1);

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

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

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

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

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

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

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

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

          rc = bcp_bind(w_hdbc1, (BYTE *) &d_next_o_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT4, 11);
          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++)
                    {
                              MakeAlphaString(6,10,D_NAME_LEN, d_name);
```

```
                                MakeAddress(d_street_1, d_street_2, d_city, d_state,
d_zip);

                                d_tax = ((float) RandomNumber(0L,2000L))/10000.00;

                                rc = bcp_sendrow(w_hdbc1);
                                if (rc != 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()
{
    long  s_i_id;
    short 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];

        // 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");

        rc = bcp_init(w_hdbc1, name, NULL, "logs\\stock.err", 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);
        }

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

        bcp_bind(w_hdbc1, (BYTE *) &s_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
2);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

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

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

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

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

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

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

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

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

```
        rc = bcp_bind(w_hdbc1, (BYTE *) s_dist_08, 0, S_DIST_LEN, NULL, 0, 0, 11);
        if (rc != SUCCEED)
                HandleErrorDBC(w_hdbc1);

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

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

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

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

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

        rc = bcp_bind(w_hdbc1, (BYTE *) s_data, 0, S_DATA_LEN, NULL, 0, 0, 17);
        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 = (short)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(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;
        short                   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];
        // SQLRETURN                                    rc_1;
        // SQLSMALLINT                                  recnum, MsgLen;
        // SQLCHAR                                              SqlState[6],
Msg[SQL_MAX_MESSAGE_LENGTH];
        // SQLINTEGER                                   NativeError;

        // 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");

        // Initialize bulk copy
        sprintf(name, "%s..%s", aptr->database, "customer");

        rc = bcp_init(c_hdbc1, name, NULL, "logs\\customer.err", 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);
        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 = (short)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");

        // 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, "isql -S%s -U%s -P%s -d%s -e -Q\"update customer set c_first
= 'C_LOAD = %d' where c_id = 1 and c_w_id = 1 and c_d_id = 1\" >
logs\\nurand_load.log",
                                        aptr->server,
                                        aptr->user,
                                        aptr->password,
                                        aptr->database,
                                        LOADER_NURAND_C);

        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()
{
        int     i;

        for (i=0;i<customers_per_district;i++)
        {
                customer_buf[i].c_id = 0;
                customer_buf[i].c_d_id = 0;
                customer_buf[i].c_w_id = 0;

                strcpy(customer_buf[i].c_first,"");
                strcpy(customer_buf[i].c_middle,"");
                strcpy(customer_buf[i].c_last,"");
                strcpy(customer_buf[i].c_street_1,"");
                strcpy(customer_buf[i].c_street_2,"");
                strcpy(customer_buf[i].c_city,"");
                strcpy(customer_buf[i].c_state,"");
                strcpy(customer_buf[i].c_zip,"");
                strcpy(customer_buf[i].c_phone,"");
                strcpy(customer_buf[i].c_credit,"");

                customer_buf[i].c_credit_lim = 0;
                customer_buf[i].c_discount = (float) 0;

                //  fix to avoid ODBC float to numeric conversion problem.
                //       customer_buf[i].c_balance = 0;
                strcpy(customer_buf[i].c_balance,"");
                customer_buf[i].c_ytd_payment = 0;
                customer_buf[i].c_payment_cnt = 0;
                customer_buf[i].c_delivery_cnt = 0;

                strcpy(customer_buf[i].c_data,"");

                customer_buf[i].h_amount = 0;

                strcpy(customer_buf[i].h_data,"");

        }


}


//========================================================================
//
// Function   : CustomerBufLoad
//
// Fills shared buffer for HISTORY and CUSTOMER
//========================================================================

void CustomerBufLoad(int d_id, int w_id)
{
        long                                    i;
    CUSTOMER_SORT_STRUCT    c[CUSTOMERS_PER_DISTRICT];


        for (i=0;i<customers_per_district;i++)
        {
                if (i < 1000)
                        LastName(i, c[i].c_last);
                else
                        LastName(NURand(255,0,999,LOADER_NURAND_C),
c[i].c_last);

                MakeAlphaString(8,16,FIRST_NAME_LEN, c[i].c_first);

                c[i].c_id = i+1;

        }

        printf("...Loading customer buffer for: d_id = %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;

                // Generate CUSTOMER and HISTORY data

                customer_buf[i].c_id = c[i].c_id;

                strcpy(customer_buf[i].c_first, c[i].c_first);
```

```
                    strcpy(customer_buf[i].c_last, c[i].c_last);

                    customer_buf[i].c_middle[0] = 'O';
                    customer_buf[i].c_middle[1] = 'E';

                    MakeAddress(customer_buf[i].c_street_1,
                                customer_buf[i].c_street_2,
                                customer_buf[i].c_city,
                                customer_buf[i].c_state,
                                customer_buf[i].c_zip);

                    MakeNumberString(16, 16, PHONE_LEN, customer_buf[i].c_phone);

                    if (RandomNumber(1L, 100L) > 10)
                            customer_buf[i].c_credit[0] = 'G';
                    else
                            customer_buf[i].c_credit[0] = 'B';
                    customer_buf[i].c_credit[1] = 'C';

                    customer_buf[i].c_credit_lim = 50000.0;
                    customer_buf[i].c_discount = ((float) RandomNumber(0L, 5000L)) /
10000.0;

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

                    //  customer_buf[i].c_balance = -10.0;
                    strcpy(customer_buf[i].c_balance,"-10.0");

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


                    // Generate HISTORY data
                    MakeAlphaString(12, 24, H_DATA_LEN, customer_buf[i].h_data);

            }
}

//============================================================================
//
// Function   : LoadCustomerTable
//
//============================================================================

void LoadCustomerTable(LOADER_TIME_STRUCT *customer_time_start)
{
            int         i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN+1];
    char        c_last[LAST_NAME_LEN+1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];
    char        c_zip[ZIP_LEN+1];
    char        c_phone[PHONE_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_credit_lim;
    double      c_discount;

            //  fix to avoid ODBC float to numeric conversion problem.
            //  double                  c_balance;
            char                c_balance[6];

    double      c_ytd_payment;
    short       c_payment_cnt;
    short       c_delivery_cnt;
    char        c_data[C_DATA_LEN+1];
            char                c_since[C_SINCE_LEN+1];
            RETCODE             rc;

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

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

            rc = bcp_bind(c_hdbc1, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 3);
            if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc1);

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

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

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

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

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

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

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

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

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

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

```
                HandleErrorDBC(c_hdbc1);

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

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

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


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

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


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

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

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

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

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

                //  c_balance = customer_buf[i].c_balance;
                strcpy(c_balance, customer_buf[i].c_balance);

                c_ytd_payment = customer_buf[i].c_ytd_payment;
                c_payment_cnt = customer_buf[i].c_payment_cnt;
                c_delivery_cnt = customer_buf[i].c_delivery_cnt;

                strcpy(c_data, customer_buf[i].c_data);

                // Send data to server
                rc = bcp_sendrow(c_hdbc1);
                if (rc != 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)
{
         int         i;
    long        c_id;
    short       c_d_id;
    short       c_w_id;
         double    h_amount;
    char        h_data[H_DATA_LEN+1];
         char               h_date[H_DATE_LEN+1];
         RETCODE            rc;

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

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

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
         if (rc != SUCCEED)
                HandleErrorDBC(c_hdbc2);
```

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

    rc = bcp_bind(c_hdbc2, (BYTE *) &c_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
5);
          if (rc != SUCCEED)
                    HandleErrorDBC(c_hdbc2);

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

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

    rc = bcp_bind(c_hdbc2, (BYTE *) h_data, 0, H_DATA_LEN, NULL, 0, 0, 8);
          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;
          short                        w_id;
    short                    d_id;
          DWORD                        dwThreadID[MAX_ORDER_THREADS];
          HANDLE                       hThread[MAX_ORDER_THREADS];
          char                         name[20];
          RETCODE                                             rc;
          char                                                bcphint[128];

          // 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);
          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);
          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);
          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 = (short)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(int d_id, int w_id)
{

         int     cust[ORDERS_PER_DISTRICT+1];
         long    o_id;
         short   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;
        short     o_w_id;
    long        o_c_id;
    short       o_carrier_id;
    short       o_ol_cnt;
    short       o_all_local;
        char              o_entry_d[O_ENTRY_D_LEN+1];
        RETCODE           rc;
        DBINT             rcint;

        // bind ORDER data
    rc = bcp_bind(o_hdbc1, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

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

    rc = bcp_bind(o_hdbc1, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
        if (rc != SUCCEED)
                HandleErrorDBC(o_hdbc1);

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

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

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

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

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

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

        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)
{
    int         i;
    long        o_id;
    short       o_d_id;
    short       o_w_id;
        RETCODE           rc;
        DBINT             rcint;

        // Bind NEW-ORDER data

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

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

```
    rc = bcp_bind(o_hdbc2, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
         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(o_hdbc2, o_hstmt2, new_order_rows_loaded,
"new_order", &new_order_time_start->time_start);
         }

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

         if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
         {
                  rcint = bcp_done(o_hdbc2);
                  if (rcint < 0)
                           HandleErrorDBC(o_hdbc2);

                  SQLFreeStmt(o_hstmt2, SQL_DROP);
                  SQLDisconnect(o_hdbc2);
                  SQLFreeConnect(o_hdbc2);

                  // if build index after load...
                  if ((aptr->build_index == 1) && (aptr->index_order == 0))
                           BuildIndex("idxnodcl");

         }

}


//============================================================================
//
// Function   : LoadOrderLineTable
//
//============================================================================

void LoadOrderLineTable(LOADER_TIME_STRUCT *order_line_time_start)
{
         int         i,j;
    long        o_id;
         short     o_d_id;
         short     o_w_id;
    long        ol;
         long      ol_i_id;
    short       ol_supply_w_id;
    short       ol_quantity;
    double      ol_amount;
    char        ol_dist_info[DIST_INFO_LEN+1];
         char                ol_delivery_d[OL_DELIVERY_D_LEN+1];
         RETCODE             rc;
         DBINT               rcint;

         // bind ORDER-LINE data
    rc = bcp_bind(o_hdbc3, (BYTE *) &o_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 1);
         if (rc != SUCCEED)
                  HandleErrorDBC(o_hdbc3);

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

    rc = bcp_bind(o_hdbc3, (BYTE *) &o_w_id, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT2,
3);
         if (rc != SUCCEED)
                  HandleErrorDBC(o_hdbc3);

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol, 0, SQL_VARLEN_DATA, NULL, 0, SQLINT4, 4);
         if (rc != SUCCEED)
                  HandleErrorDBC(o_hdbc3);

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

    rc = bcp_bind(o_hdbc3, (BYTE *) &ol_supply_w_id, 0, SQL_VARLEN_DATA, NULL, 0,
SQLINT2, 6);
         if (rc != SUCCEED)
                  HandleErrorDBC(o_hdbc3);

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

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

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

    rc = bcp_bind(o_hdbc3, (BYTE *) ol_dist_info, 0, DIST_INFO_LEN, NULL, 0, 0, 10);
         if (rc != SUCCEED)
                  HandleErrorDBC(o_hdbc3);

         for (i = 0; i < orders_per_district; i++)
    {
                  o_id    = orders_buf[i].o_id;
                  o_d_id  = orders_buf[i].o_d_id;
                  o_w_id  = orders_buf[i].o_w_id;

                  for (j=0; j < orders_buf[i].o_ol_cnt; j++)
                  {
                           ol             = orders_buf[i].o_ol[j].ol;
                           ol_i_id        = orders_buf[i].o_ol[j].ol_i_id;
                           ol_supply_w_id = orders_buf[i].o_ol[j].ol_supply_w_id;
                           ol_quantity    = orders_buf[i].o_ol[j].ol_quantity;
```

```
                        ol_amount      = orders_buf[i].o_ol[j].ol_amount;


        strcpy(ol_delivery_d,orders_buf[i].o_ol[j].ol_delivery_d);


        strcpy(ol_dist_info,orders_buf[i].o_ol[j].ol_dist_info);

                        rc = bcp_sendrow(o_hdbc3);
                        if (rc != SUCCEED)
                                HandleErrorDBC(o_hdbc3);

                        order_line_rows_loaded++;
                        CheckForCommit(o_hdbc3, o_hstmt3,
order_line_rows_loaded, "order_line", &order_line_time_start->time_start);
                }

        }

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

        if ((o_w_id == aptr->num_warehouses) && (o_d_id == 10))
        {
                rcint = bcp_done(o_hdbc3);
                if (rcint < 0)
                        HandleErrorDBC(o_hdbc3);

                SQLFreeStmt(o_hstmt3, SQL_DROP);
                SQLDisconnect(o_hdbc3);
                SQLFreeConnect(o_hdbc3);

                // if build index after load...
                if ((aptr->build_index == 1) && (aptr->index_order == 0))
                        BuildIndex("idxodlcl");

        }

}


//=============================================================================
//
// Function   : GetPermutation
//
//=============================================================================

void GetPermutation(int perm[], int n)
{
        int i, r, t;

        for (i=1;i<=n;i++)
                perm[i] = i;

        for (i=1;i<=n;i++)
        {
                r = RandomNumber(i,n);
                t = perm[i];
                perm[i] = perm[r];
                perm[r] = t;
        }
}


//=============================================================================
//
// Function   : CheckForCommit
//
//=============================================================================

void CheckForCommit(HDBC hdbc,
                                                HSTMT hstmt,
                                        int rows_loaded,
                                                char *table_name,
                                        long *time_start)
{

    long  time_end, time_diff;
          // DBINT  rcint;

    if ( !(rows_loaded % aptr->batch) )
    {

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

                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   : 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)
                 HandleErrorDBC(i_hdbc1);

        // 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)
                 HandleErrorDBC(w_hdbc1);

        // 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)
                 HandleErrorDBC(c_hdbc1);

        // 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)
                HandleErrorDBC(c_hdbc2);

        // 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)
                HandleErrorDBC(o_hdbc1);

        // 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)
                HandleErrorDBC(o_hdbc2);

        // 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)
                HandleErrorDBC(o_hdbc3);

}

//============================================================================
//
// Function name: BuildIndex
//
//============================================================================

void BuildIndex(char         *index_script)
{
        char     cmd[256];

        printf("Starting index creation:  %s\n",index_script);

        sprintf(cmd, "isql -S%s -U%s -P%s -e -i%s\\%s.sql > logs\\%s.log",
                                aptr->server,
                                aptr->user,
                                aptr->password,
                                aptr->index_script_path,
                                index_script,
                                index_script);

        system(cmd);
```

```
        printf("Finished index creation:  %s\n",index_script);
}


void HandleErrorDBC (SQLHDBC  hdbc1)
{

        SQLCHAR           SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char              timebuf[128];
        char              datebuf[128];
        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" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}

void HandleErrorSTMT (HSTMT  hstmt1)
{

        SQLCHAR           SqlState[6], Msg[SQL_MAX_MESSAGE_LENGTH];
        SQLINTEGER  NativeError;
        SQLSMALLINT i, MsgLen;
        SQLRETURN   rc2;
        char              timebuf[128];
        char              datebuf[128];
        FILE              *fp1;

        i = 1;
        while (( rc2 = SQLGetDiagRec(SQL_HANDLE_STMT , hstmt1, i, SqlState ,
&NativeError,
                                    Msg, sizeof(Msg) , &MsgLen )) !=
SQL_NO_DATA )
        {
```

```
                sprintf( szLastError , "%s" ,  Msg );

                _strtime(timebuf);
                _strdate(datebuf);

                printf( "[%s : %s] %s\n" , datebuf, timebuf, szLastError);

                fp1 = fopen("logs\\tpccldr.err","w");
                if (fp1 == NULL)
                        printf("ERROR:  Unable to open errorlog file.\n");
                else
                {
                        fprintf(fp1, "[%s : %s] %s\n" , datebuf, timebuf,
szLastError);
                        fclose(fp1);
                }

                i++;

        }
}


void FormatDate ( char* szTimeCOutput )
{

        struct tm when;
        time_t now;

        time( &now );
        when  = *localtime( &now );

        mktime( &when );

        // odbc datetime format
        strftime( szTimeCOutput , 30 , "%Y-%m-%d %H:%M:%S.000", &when );

        return;

}

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

void CheckDataBase()
{

        RETCODE         rc;

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

        SQLSMALLINT                 cbDriverStringOut;
        SQLCHAR                     TabName[10];
        SQLINTEGER                  TabNameInd, TabCount, TabCountInd;


        ExitFlag = 0;
```

```
        SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv );

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

        SQLAllocHandle(SQL_HANDLE_DBC, henv , &v_hdbc);

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

        // Open connection to SQL Server

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


        rc = SQLSetConnectAttr( v_hdbc, SQL_ATTR_PACKET_SIZE, (SQLPOINTER)aptr-
>pack_size, SQL_IS_UINTEGER );
        if (rc != SQL_SUCCESS)
                HandleErrorDBC(v_hdbc);

        rc = SQLDriverConnect ( v_hdbc,
                                                NULL,

        (SQLCHAR*)&szDriverString[0] ,
                                                        SQL_NTS,

        (SQLCHAR*)&szDriverStringOut[0],

        sizeof(szDriverStringOut),
                                                        &cbDriverStringOut,
                                                        SQL_DRIVER_NOPROMPT );

        // if the rc is SQL_ERROR, the the TPCC database probably does not exist
        if (rc == SQL_ERROR)
        {
                printf("The database TPCC does not appear to exist!\n");
                printf("\nCheck LOGS\\ directory for database creation
errors.\n");

                // cleanup database connections and handles
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                SQLDisconnect(v_hdbc);
                SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                // since there is not a database, exit back to SETUP.CMD
                exit(1);
        }

        if ( SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt) != SQL_SUCCESS )
                HandleErrorDBC(v_hdbc);

        if ( SQLBindCol(v_hstmt, 1, SQL_C_ULONG, &TabCount, 0, &TabCountInd) !=
SQL_SUCCESS )
                HandleErrorSTMT(v_hstmt);

        // count the number of user tables from sysobjects
        rc = SQLExecDirect(v_hstmt, "select count(*) from sysobjects where xtype =
\'U\'", SQL_NTS);
        if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                HandleErrorSTMT(v_hstmt);

        if ( SQLFetch(v_hstmt) != SQL_SUCCESS )
                HandleErrorSTMT(v_hstmt);

        // if the number of tables is less than 9, select all the user tables in
TPCC
        if (TabCount != 9)
        {
                SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);

                SQLAllocHandle(SQL_HANDLE_STMT, v_hdbc , &v_hstmt);

                if ( SQLBindCol(v_hstmt, 1, SQL_C_CHAR, &TabName,
sizeof(TabName), &TabNameInd) != SQL_SUCCESS )
                        HandleErrorSTMT(v_hstmt);

                // select the list of user tables into a result set
                rc = SQLExecDirect(v_hstmt, "select * from sysobjects where
xtype = \'U\'", SQL_NTS);
                if ((rc != SQL_SUCCESS) && (rc != SQL_SUCCESS_WITH_INFO))
                        HandleErrorSTMT(v_hstmt);

                // go through the result set and set the bitmap for each found
table
                // set the bitmap to '1' if the table name is found

                while ((rc = SQLFetch(v_hstmt)) != SQL_NO_DATA)
                {
                        switch( TabName[0] )
                        {
                        case 'w':
                                TablesBitMap[0] = '1';
                                break;
                        case 'd':
                                TablesBitMap[1] = '1';
                                break;
                        case 'c':
                                TablesBitMap[2] = '1';
                                break;
                        case 'h':
                                TablesBitMap[3] = '1';
                                break;
                        case 'n':
                                TablesBitMap[4] = '1';
                                break;
                        case 'o':
                                if (TabName[5] = 's')
                                        TablesBitMap[5] = '1';
                                if (TabName[5] = '_')
                                        TablesBitMap[6] = '1';
                                break;
                        case 'i':
                                TablesBitMap[7] = '1';
                                break;
                        case 's':
                                TablesBitMap[8] = '1';
                                break;
                        }
                }

                // a '0' ExitFlag means do NOT exit the loader early, a '1'
means exit the loader early
```

```
                ExitFlag = 0;

                // interate through the bitmap to display which table(s) is
actually missing
                for (i = 0; i <= 8; i++)
                {
                        switch(i)
                        {
                        case 0:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Warehouse table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 1:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The District table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 2:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Customer table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 3:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The History table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 4:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The New_Order table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 5:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Orders table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 6:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Order_Line table is
missing or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 7:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Item table is missing
or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        case 8:
                                if (TablesBitMap[i] == '0')
                                {
                                        printf("The Stock table is missing
or damaged.\n");
                                        ExitFlag = 1;
                                }
                                break;
                        }
                }

                // if one or more tables are missing, display message and exit
the loader
                if (ExitFlag = 1)
                {
                        printf("\nExiting TPC-C Loader!\n");
                        printf("\nCheck LOGS\\ directory for database\n");
                        printf("or table creation errors.\n");

                        // cleanup database connections and handles
                        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
                        SQLDisconnect(v_hdbc);
                        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

                        exit(1);
                }
        }

        // cleanup database connections and handles
        SQLFreeHandle(SQL_HANDLE_STMT, v_hstmt);
        SQLDisconnect(v_hdbc);
        SQLFreeHandle(SQL_HANDLE_DBC, v_hdbc);

        return;
}
```

verify_msg.sql

```
exec sp_dropmessage 50003
exec sp_addmessage 50003, 1,"Incorrect Sort Order - Please re-install SQL Server
with the Binary Sort Order"
```

verify_sort.sql

```
--  File:     VERIFY_SORT.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Verifies the Sort Order

if exists (select name from sysobjects where name = "ms_verify_sort" )
        drop procedure ms_verify_sort
go
```

```
create proc ms_verify_sort

as

declare   @sort_order        int


-- get the sort order
select    @sort_order        = (select value from sysconfigures where config =
'1123')

if (select @sort_order) <> 50
         RAISERROR (50003,11,1)
go
```

version.sql

```
--  File:     VERSION.SQL
--            Microsoft TPC-C Benchmark Kit Ver. 4.21.000
--            Copyright Microsoft, 1999, 2000
--  Purpose:  Returns version level of TPC-C stored procs
--  Note:     Always update the return value of this proc for
--            any interface changes or "must have" bug fixes.
--
--  The value returned by this SP defines the "interface level",
--  which must match between the stored procs and the client code.
--  The interface level may be down rev from the current kit.  This
--  indicates that the interface hasn't changed since that version.

use tpcc
go

if exists ( select name from sysobjects where name = "tpcc_version" )
         drop procedure tpcc_version
go

create proc tpcc_version
as
declare   @version  char(8)

begin
         select @version = "4.10.000"
         select @version as "Version"
end

go
```

Appendix C: Tunable Parameters

Microsoft SQL Server 8.0 Startup Parameters

C:\SQL80\MSSQL\BINN\SQLSERVR
-eC:\MSSQL7\LOG\ERRORLOG -x -c -t3502 –g100

Where:
-c Start SQL Server independently of the Windows NT Service Control Manager
-x Disables the keeping of CPU time and cache-hit ratio statistics
-t3502 Prints a message to the SQL Server log at the start and end of each checkpoint
-g100 Specify the amount of virtual address space in MB, SQL Server will leave available for memory allocations, excluding the buffer pool and threads stack, such as dynamically-loaded DLLs, extended procedure calls, etc. Incorrect use of this option can lead to conditions under which SQL Server may not start or may encounter runtime errors.

Microsoft SQL Server 2000 Configuration Parameters

name	minimum	maximum	config_value	run_value
affinity mask	0	2147483647	7	7
allow updates	0	1	0	0
awe enabled	0	1	1	1
c2 audit mode	0	1	0	0
cost threshold for parallelism	0	32767	5	5
cursor threshold	-1	2147483647	-1	-1
default full-text language	0	2147483647	1033	1033
default language	0	9999	0	0
fill factor (%)	0	100	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	32	1	1
max server memory (MB)	4	2147483647	1800	1800
max text repl size (B)	0	2147483647	65536	65536
max worker threads	32	32767	350	350
media retention	0	365	0	0
min memory per query (KB)	512	2147483647	1024	1024
min server memory (MB)	0	2147483647	1800	1800
nested triggers	0	1	1	1
network packet size (B)	512	65536	4096	4096
open objects	0	2147483647	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	35	35
remote access	0	1	1	1
remote login timeout (s)	0	2147483647	20	20
remote proc trans	0	1	0	0
remote query timeout (s)	0	2147483647	600	600
scan for startup procs	0	1	0	0
set working set size	0	1	1	1
show advanced options	0	1	1	1
two digit year cutoff	1753	9999	2049	2049
user connections	0	32767	360	360

```
 user options                                                                0
32767           0           0

1>
```

Benchcraft Profile

```
Profile:  cheapo
File Path:          G:\cheapo.pro
Version:  2

          Number of Engines: 2

                    Name: N13
                    Description:
                    Directory: c:\bench.log
                    Machine: N13
                    Parameter Set: 2.0 Think Time
                    Index: 0
                    Seed: 33611
                    Configured Users: 8100
                    Pipe Name: DRIVER159717062
                    Connect Rate: 500
                    Start Rate: 0
                    Max. Concurrency: 0
                    Concurrency Rate: 0
                    CLIENT_NURAND: 233
                    CPU: 0

                    Name: N14
                    Description:
                    Directory: c:\bench.log
                    Machine: N14
                    Parameter Set: 2.0 Think Time
                    Index: 100000000
                    Seed: 33611
                    Configured Users: 8100
                    Pipe Name: DRIVER259763609
                    Connect Rate: 500
                    Start Rate: 0
                    Max. Concurrency: 0
                    Concurrency Rate: 0
                    CLIENT_NURAND: 233
                    CPU: 0

          Number of User groups: 2

                    Driver Engine: N13
                    IIS Server: FE37C
                    SQL Server: cheapo_ip
                    Database: tpcc
                    User: sa
                    Protocol: HTML
                    w_id Range: 1 - 810
                    w_id Min Warehouse: 1
                    w_id Max Warehouse: 1620
                    Scale: Normal
                    User Count: 8100
                    District id: 1
                    Scale Down: No

                    Driver Engine: N14
                    IIS Server: FE38C
                    SQL Server: cheapo_ip
                    Database: tpcc
                    User: sa
                    Protocol: HTML
                    w_id Range: 811 - 1620
                    w_id Min Warehouse: 1
                    w_id Max Warehouse: 1620
                    Scale: Normal
                    User Count: 8100
                    District id: 1
                    Scale Down: No

          Number of Parameter Sets: 11

                    ~Default
                    Default Parameter Set
                                        Txn     Think   Key     RT      RT      Menu
                                        Weight  Time    Time    Delay   Fence   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

                    2.0 Think Time
                                        Txn     Think   Key     RT      RT      Menu
                                        Weight  Time    Time    Delay   Fence   Delay
                    New Order             44.74         24.00    18.01     0.10
5.00        0.10
                    Payment               43.09         24.00     3.01     0.10
5.00        0.10
                    Delivery               4.05         10.00     2.01     0.10
5.00        0.10
                    Stock Level            4.07         10.00     2.01     0.10
20.00       0.10
                    Order Status           4.05         20.00     2.01     0.10
5.00        0.10

                    1.5 Think Time
                                        Txn     Think   Key     RT      RT      Menu
                                        Weight  Time    Time    Delay   Fence   Delay
                    New Order             44.74         18.00    18.01     0.10
5.00        0.10
                    Payment               43.09         18.00     3.01     0.10
5.00        0.10
                    Delivery               4.05          7.50     2.01     0.10
5.00        0.10
                    Stock Level            4.07          7.50     2.01     0.10
20.00       0.10
                    Order Status           4.05         15.00     2.01     0.10
5.00        0.10

                    1.1 Think Time
```

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		13.20	18.01	0.10	0.10
Payment	5.00	43.09		13.20	3.01	0.10	0.10
Delivery	5.00	4.05		5.50	2.01	0.10	0.10
Stock Level	20.00	4.07		5.50	2.01	0.10	0.10
Order Status	5.00	4.05		11.00	2.01	0.10	0.10

1.08 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		12.96	18.01	0.10	0.10
Payment	5.00	43.09		12.96	3.01	0.10	0.10
Delivery	5.00	4.05		5.40	2.01	0.10	0.10
Stock Level	20.00	4.07		5.40	2.01	0.10	0.10
Order Status	5.00	4.05		10.80	2.01	0.10	0.10

1.06 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		12.72	18.01	0.10	0.10
Payment	5.00	43.09		12.72	3.01	0.10	0.10
Delivery	5.00	4.05		5.30	2.01	0.10	0.10
Stock Level	20.00	4.07		5.30	2.01	0.10	0.10
Order Status	5.00	4.05		10.60	2.01	0.10	0.10

1.03 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		12.36	18.01	0.10	0.10
Payment	5.00	43.09		12.36	3.01	0.10	0.10
Delivery	5.00	4.05		5.15	2.01	0.10	0.10
Stock Level	20.00	4.07		5.15	2.01	0.10	0.10
Order Status	5.00	4.05		10.30	2.01	0.10	0.10

1.02 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		12.24	18.01	0.10	0.10
Payment	5.00	43.09		12.24	3.01	0.10	0.10
Delivery	5.00	4.05		5.10	2.01	0.10	0.10
Stock Level	20.00	4.07		5.10	2.01	0.10	0.10
Order Status	5.00	4.05		10.20	2.01	0.10	0.10

1.01 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		12.12	18.01	0.10	0.10
Payment	5.00	43.09		12.12	3.01	0.10	0.10
Delivery	5.00	4.05		5.05	2.01	0.10	0.10
Stock Level	20.00	4.07		5.05	2.01	0.10	0.10
Order Status	5.00	4.05		10.10	2.01	0.10	0.10

2.5 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		30.00	18.01	0.10	0.10
Payment	5.00	43.09		30.00	3.01	0.10	0.10
Delivery	5.00	4.05		12.50	2.01	0.10	0.10
Stock Level	20.00	4.07		12.50	2.01	0.10	0.10
Order Status	5.00	4.05		25.00	2.01	0.10	0.10

3.0 Think Time

		Txn Weight	Think Time	Key Time	RT Delay	RT Fence	Menu Delay
New Order	5.00	44.74		36.00	18.01	0.10	0.10
Payment	5.00	43.09		35.00	3.01	0.10	0.10
Delivery	5.00	4.05		15.00	2.01	0.10	0.10
Stock Level	20.00	4.07		15.00	2.01	0.10	0.10
Order Status	5.00	4.05		30.00	2.01	0.10	0.10

Internet Information Server Registry Parameters

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:00000019
"DispatchEntries"=hex(7):4c,44,41,50,53,56,43,00,00
"PoolThreadLimit"=dword:000003fe
"ThreadTimeout"=dword:00015180

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.dll"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000842
"Last Help"=dword:00000843
"First Counter"=dword:00000802
"First Help"=dword:00000803
"Library Validation Code"=hex:f6,7c,05,e6,c3,f8,bf,01,10,25,00,00,00,00,00,00
```

World Wide Web Service Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,69,6e,\
  65,74,73,72,76,5c,69,6e,65,74,69,6e,66,6f,2e,65,78,65,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):49,49,53,41,44,4d,49,4e,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity and administration through the Internet
Information Services snap-in."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility only."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\ASP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"CertMapList"="C:\\WINNT\\System32\\inetsrv\\iiscrmap.dll"
"AccessDeniedMessage"="Error: Access is Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\LogFiles"
"AcceptExOutstanding"=dword:00000028

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\Adv
ancedDataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDS
Server.DataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual
Roots]
"/"="c:\\inetpub\\wwwroot,,205"
"/Scripts"="c:\\inetpub\\scripts,,204"
"/IISHelp"="c:\\winnt\\help\\iishelp,,201"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,201"
"/IISSamples"="c:\\inetpub\\iissamples,,201"
"/MSADC"="c:\\program files\\common files\\system\\msadc,,205"
"/Printers"="C:\\WINNT\\web\\printers,,201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation Code"=hex:c8,20,d3,e8,c3,f8,bf,01,10,3d,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,72,00,73,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Server Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Abiosdsk]
"ErrorControl"=dword:00000000
"Group"="Primary disk"
"Start"=dword:00000004
"Tag"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\abp480n5]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000038
"Type"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\abp480n5\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\abp480n5\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ACPI]
"ErrorControl"=dword:00000001
"Group"="Boot Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000001
"Type"=dword:00000001
"DisplayName"="Microsoft ACPI Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,41,43,50,\
  49,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ACPI\Enum]
"0"="ACPI_HAL\\PNP0C08\\0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ACPIEC]
"ErrorControl"=dword:00000001
"Group"="Boot Bus Extender"
"Start"=dword:00000004
"Tag"=dword:00000005
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\adpu160m]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000003c
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\adpu160m\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\adpu160m\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD]
"Type"=dword:00000001
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,61,66,64,2e,73,79,73,00
"DisplayName"="AFD Networking Support Environment"
"Group"="TDI"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AFD\Enum]
"0"="Root\\LEGACY_AFD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Aha154x]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000006
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Aha154x\Parameters]
"LegacyAdapterDetection"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Aha154x\Parameters\PnpInterfac
e]
"1"=dword:00000001
"3"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic116x]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000033
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic116x\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic116x\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78u2]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000034
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78u2\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78u2\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78xx]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000001e
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78xx\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\aic78xx\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Alerter]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Alerter"
```

```
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Notifies selected users and computers of administrative alerts."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Alerter\Parameters]
"AlertNames"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Alerter\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Alerter\Enum]
"0"="Root\\LEGACY_ALERTER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ami0nt]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000000d
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\amsint]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000024
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\amsint\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\amsint\Parameters\PnpInterface
]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AppMgmt]
"Description"="Provides software installation services such as Assign, Publish, and
Remove."
"DisplayName"="Application Management"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AppMgmt\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,03,\
  18,00,9d,00,00,00,01,02,00,00,00,00,00,05,20,00,00,00,21,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AppMgmt\Enum]
"0"="Root\\LEGACY_APPMGMT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000029
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc\Parameters\PnpInterface]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3350p]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000039
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3350p\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3350p\Parameters\PnpInterfa
ce]
"1"=dword:00000011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3550]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000002a
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3550\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\asc3550\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AsyncMac]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,61,73,79,\
  6e,63,6d,61,63,2e,73,79,73,00
"DisplayName"="RAS Asynchronous Media Driver"
"Description"="RAS Asynchronous Media Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AsyncMac\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6c,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atapi]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:00000019
"Type"=dword:00000001
"DisplayName"="Standard IDE/ESDI Hard Disk Controller"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,61,74,61,\
  70,69,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atapi\Parameters]
"LegacyDetection"=dword:00000001
"GhostSlave"=hex(7):53,75,6e,44,69,73,6b,20,00,00
"UseCheckPowerForFlush"=hex(7):53,41,4d,53,55,4e,47,20,57,4e,52,2d,33,31,36,30,\
  31,41,20,28,31,36,30,30,4d,42,29,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,\
  53,41,4d,53,55,4e,47,20,57,4e,52,2d,33,31,36,30,31,41,20,28,31,2e,36,47,42,\
  29,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,49,42,4d,2d,44,54,43,41,2d,\
  32,34,30,39,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,54,43,36,4f,41,41,32,41,00,49,42,4d,2d,44,54,43,41,2d,32,\
  34,30,39,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,54,43,36,49,41,41,32,41,00,49,42,4d,2d,44,50,4c,41,2d,32,35,\
  31,32,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,50,4c,38,4f,41,41,32,41,00,49,42,4d,2d,44,50,4c,41,2d,32,35,31,\
  32,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,50,4c,38,49,41,41,32,41,00,49,42,4d,2d,44,50,4c,41,2d,32,35,31,32,\
  30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,50,4c,38,49,41,41,34,41,00,49,42,4d,2d,44,54,43,41,2d,32,33,32,34,30,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,54,43,35,4f,41,41,32,41,00,49,42,4d,2d,44,54,43,41,2d,32,33,32,34,30,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  54,43,35,49,41,41,32,41,00,49,42,4d,2d,44,50,4c,41,2d,32,34,34,38,30,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,50,\
  4c,37,4f,41,41,32,41,00,49,42,4d,2d,44,50,4c,41,2d,32,34,34,38,30,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,50,4c,\
  37,49,41,41,32,41,00,00
"NoFlushDevice"=hex(7):51,55,41,4e,54,55,4d,5f,4c,50,53,35,32,35,41,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,53,43,\
  52,2d,37,33,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,00
"PioOnlyDevice"=hex(7):20,20,20,20,43,6f,6e,6e,65,72,20,50,65,72,69,70,68,65,\
  72,61,6c,73,20,34,32,35,4d,42,20,2d,20,43,46,53,34,32,35,41,20,20,00,4d,41,\
  54,53,48,49,54,41,20,43,52,2d,35,38,31,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,46,58,36,30,\
  30,53,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,00,43,44,2d,34,34,45,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,51,55,41,4e,\
  54,55,4d,20,54,52,42,38,35,30,41,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,00,51,55,41,4e,54,55,4d,20,4d,41,52,56,45,\
  52,49,43,4b,20,35,34,30,41,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,00,20,4d,41,58,54,4f,52,20,4d,58,54,2d,35,34,30,20,20,41,54,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,\
  20,37,31,32,36,30,20,41,54,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,20,37,38,35,30,20,41,56,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  00,4d,61,78,74,6f,72,20,37,35,34,30,20,41,56,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,20,37,\
  32,31,33,20,41,54,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,20,37,33,34,35,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,4d,\
  61,78,74,6f,72,20,37,32,34,35,20,41,54,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,20,37,32,34,\
  35,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,00,4d,61,78,74,6f,72,20,37,32,31,31,41,55,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,4d,61,78,\
  74,6f,72,20,37,32,31,37,31,20,41,54,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,00,43,44,2d,33,31,36,45,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,00,53,41,4d,53,55,4e,47,5f,53,43,52,2d,32,34,33,30,00,43,52,2d,32,\
  38,30,31,54,45,00,00
"NonRemovableMedia"=hex(7):4b,69,6e,67,73,74,6f,6e,20,54,65,63,68,6e,6f,6c,6f,\
  67,79,20,44,61,74,61,50,61,6b,20,33,34,30,20,20,20,20,20,20,20,20,20,20,00,53,\
  75,6e,44,69,73,6b,20,53,44,50,35,41,2d,31,30,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,53,75,6e,44,69,73,6b,20,53,44,\
  43,46,42,2d,31,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,00,53,75,6e,44,69,73,6b,20,53,44,50,33,42,2d,32,30,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,53,75,6e,\
  44,69,73,6b,20,53,44,50,33,42,2d,31,37,35,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,53,75,6e,44,69,73,6b,20,53,44,50,35,\
  2d,32,2e,35,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,00,43,61,6c,6c,75,6e,61,20,54,65,63,68,6e,6f,6c,6f,67,79,20,43,54,\
  32,36,30,4d,43,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,42,4e,2d,53,30,\
  30,34,41,43,2d,53,20,31,2e,30,30,00,43,61,6c,6c,75,6e,61,20,54,65,63,68,6e,\
  6f,6c,6f,67,79,20,43,54,35,32,30,52,4d,00,48,69,74,61,63,68,69,20,43,56,20,\
  35,2e,31,2e,31,00,20,20,20,20,20,20,41,54,41,5f,46,4c,41,53,48,20,00,4d,69,\
  74,73,75,62,69,73,68,69,20,41,54,41,20,43,61,72,64,20,00,4c,45,58,41,52,20,\
  41,54,41,5f,46,4c,41,53,48,00,4d,69,63,72,6f,6e,20,4d,54,43,46,30,30,34,41,\
  00,4d,69,63,72,6f,6e,20,4d,54,43,46,30,30,38,41,00,53,75,6e,44,69,73,6b,20,\
  53,44,50,33,42,2d,31,31,30,00,53,75,6e,44,69,73,6b,20,53,44,43,46,42,2d,34,\
  00,42,4e,2d,43,41,42,2d,54,00,4d,45,4d,4f,52,59,53,54,49,43,4b,20,20,20,38,\
  4d,20,20,38,4b,00,00
"NoPowerDownDevice"=hex(7):52,44,2d,44,52,43,30,30,31,2d,4d,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,43,\
  53,2d,52,33,37,20,30,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,00,00
"AutoEjectZipDevice"=hex(7):49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,\
  20,20,20,20,20,41,54,41,50,49,20,20,20,20,20,20,20,20,20,20,20,20,20,20,32,33,\
  2e,44,20,20,20,20,20,00,49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,20,\
  20,20,20,20,41,54,41,50,49,20,20,20,20,20,20,20,20,20,20,20,20,20,20,32,31,2e,\
  44,20,20,20,20,00,49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,20,20,\
  20,20,20,41,54,41,50,49,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,32,30,2e,44,\
  20,20,20,20,00,49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,20,20,20,20,\
  20,20,41,54,41,50,49,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,39,31,2e,44,20,\
  20,20,20,00,49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,42,2e,32,39,20,20,\
  20,20,00,49,4f,4d,45,47,41,20,20,5a,49,50,20,31,30,30,20,20,20,20,20,20,20,\
  20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,42,2e,32,32,20,20,20,\
  20,00,00
"NeedIdentDevice"=hex(7):51,55,41,4e,54,55,4d,20,46,49,52,45,42,41,4c,4c,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atapi\Enum]
"0"="PCIIDE\\IDEChannel\\4&23727f60&0&0"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="PCIIDE\\IDEChannel\\4&23727f60&0&1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Atdisk]
"ErrorControl"=dword:00000000
"Group"="Primary disk"
"Start"=dword:00000004
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atirage]
"Type"=dword:00000001
"Start"=dword:00000003
```

```
"ErrorControl"=dword:00000000
"Tag"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,61,74,69,\
  72,61,67,65,6d,2e,73,79,73,00
"Group"="Video"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atirage\Device0]
"InstalledDisplayDrivers"=hex(7):61,74,69,72,61,67,65,64,00,00
"VgaCompatible"=dword:00000000
"Device Description"="ATI Technologies Inc. 3D RAGE IIC PCI"
"HardwareInformation.ChipType"=hex:41,00,54,00,49,00,20,00,33,00,44,00,20,00,\
  52,00,41,00,47,00,45,00,20,00,49,00,49,00,43,00,20,00,50,00,43,00,49,00,20,\
  00,28,00,41,00,32,00,31,00,29,00,00,00
"HardwareInformation.DacType"=hex:41,00,54,00,49,00,20,00,49,00,6e,00,74,00,65,\
  00,72,00,6e,00,61,00,6c,00,20,00,44,00,41,00,43,00,00,00
"HardwareInformation.MemorySize"=hex:00,00,40,00
"HardwareInformation.AdapterString"=hex:31,00,30,00,30,00,32,00,2d,00,34,00,37,\
  00,35,00,36,00,2d,00,37,00,41,00,2d,00,31,00,30,00,30,00,32,00,2d,00,34,00,\
  37,00,35,00,36,00,00,00
"HardwareInformation.BiosString"=hex:41,00,54,00,49,00,20,00,56,00,69,00,64,00,\
  65,00,6f,00,20,00,42,00,49,00,4f,00,53,00,00,00
"WDMConfiguration.VideoFeature.TunerType"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.DecoderType"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.DecoderAndCrystalsType"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.AudioChipType"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.OEMId"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.ATIProductId"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.ProductRevisionId"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.ConnectorsAndPassThrough"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.VoltageAndTeletext"=hex:3f,3f,3f,3f
"WDMConfiguration.VideoFeature.Reserved"=hex:3f,3f,3f,3f
"WDMConfiguration.HardwareInfo.I2CMethod"=hex:0f,00,00,00
"WDMConfiguration.HardwareInfo.TVOutSupport"=hex:00,00,00,00
"WDMConfiguration.HardwareInfo.VideoPortType"=hex:3f,3f,3f,3f
"WDMConfiguration.TVOutConfiguration.TVOutInformation"=hex:ff,ff,00,00
"WDMConfiguration.FrequencyTable.ReferenceClock"=hex:80,81,da,00
"CustomCRTTimings"=hex:00
"MultiMediaHardware.I2CExpander"=hex:18
"MultiMediaHardware.TunerId"=hex:00,00,00,00
"MultiMediaHardware.DecoderId"=hex:00,00,00,00
"MultiMediaHardware.AudioId"=hex:ff,ff,ff,ff
"HardwareInformation.Crc32"=hex:56,dd,79,4e

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atirage\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,6f,00,72,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,74,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,74,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\atirage\Enum]
"0"="PCI\\VEN_1002&DEV_4756&SUBSYS_00000000&REV_7A\\3&267a616a&0&28"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Atmarpc]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:0000000b
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,61,74,6d,\
  61,72,70,63,2e,73,79,73,00
"DisplayName"="ATM ARP Client Protocol"
"Group"="NDIS"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"Description"="ATM ARP Client Protocol"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Atmarpc\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\audstub]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,61,75,64,\
  73,74,75,62,2e,73,79,73,00
"DisplayName"="Audio Stub Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\audstub\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,61,00,73,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,65,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,65,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\audstub\Enum]
"0"="Root\\MEDIA\\MS_MMACM"
"Count"=dword:00000005
"NextInstance"=dword:00000005
"1"="Root\\MEDIA\\MS_MMDRV"
"2"="Root\\MEDIA\\MS_MMMCI"
"3"="Root\\MEDIA\\MS_MMVCD"
"4"="Root\\MEDIA\\MS_MMVID"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Beep]
"ErrorControl"=dword:00000001
"Group"="Base"
"Start"=dword:00000001
"Tag"=dword:00000002
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Beep\Enum]
"0"="Root\\LEGACY_BEEP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
```

```
"DisplayName"="Computer Browser"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  4c,61,6e,6d,61,6e,53,65,72,76,65,72,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Maintains an up-to-date list of computers on your network and
supplies the list to programs that request it."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Parameters]
"IsDomainMaster"="FALSE"
"MaintainServerList"="Yes"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Enum]
"0"="Root\\LEGACY_BROWSER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BusLogic]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000001b
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BusLogic\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BusLogic\Parameters\PnpInterfa
ce]
"1"=dword:00000001
"2"=dword:00000001
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cd20xrnt]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000003a
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cd20xrnt\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cd20xrnt\Parameters\PnpInterfa
ce]
"1"=dword:00000011

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdaudio]
"ErrorControl"=dword:00000000
"Group"="Filter"
"Start"=dword:00000001
"Tag"=dword:00000006
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdaudio\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000
"INITSTARTFAILED"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdfs]
"DependOnGroup"=hex(7):53,43,53,49,20,43,44,52,4f,4d,20,43,6c,61,73,73,00,00
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000004
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdfs\Enum]
"0"="Root\\LEGACY_CDFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom]
"Autorun"=dword:00000001
"AutoRunAlwaysDisable"=hex(7):4e,45,43,20,20,20,20,20,4d,42,52,2d,37,20,20,20,\
  00,4e,45,43,20,20,20,20,20,4d,42,52,2d,37,2e,34,20,00,50,49,4f,4e,45,45,52,\
  20,43,48,41,4e,47,52,20,44,52,4d,2d,31,38,30,34,58,00,50,49,4f,4e,45,45,52,\
  20,43,44,2d,52,4f,4d,20,44,52,4d,2d,36,33,32,34,58,00,50,49,4f,4e,45,45,52,\
  20,43,44,2d,52,4f,4d,20,44,52,4d,2d,36,32,34,58,20,00,54,4f,52,69,53,41,4e,\
  20,43,44,2d,52,4f,4d,20,43,44,52,5f,43,33,36,00,00
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,6f,72,74,00,00
"ErrorControl"=dword:00000001
"Group"="SCSI CDROM Class"
"Start"=dword:00000001
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="CD-ROM Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,64,72,\
  6f,6d,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cdrom\Enum]
"0"="IDE\\CdRomCOMPAQ_XM-6402B_____________________1723____\\5&326853dd&0&0.0.0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Changer]
"ErrorControl"=dword:00000000
"Group"="Filter"
"Start"=dword:00000001
"Tag"=dword:00000005
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cisvc]
"Type"=dword:00000120
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,63,69,\
  73,76,63,2e,65,78,65,00
"DisplayName"="Indexing Service"
"DependOnService"=hex(7):52,50,43,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cisvc\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
```

```
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ClipSrv]
"DependOnService"=hex(7):4e,65,74,44,44,45,00,00
"Description"="Supports ClipBook Viewer, which allows pages to be seen by remote
ClipBooks."
"DisplayName"="ClipBook"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,63,6c,69,70,73,72,76,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ClipSrv\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,00,\
  18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentFilter]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentFilter\Linkage]
"Bind"="\\Dummy"
"Export"="\\Dummy"
"Route"="\\Dummy"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentFilter\Performance]
"Close"="DoneFILTERPerformanceData"
"Collect"="CollectFILTERPerformanceData"
"Open"="InitializeFILTERPerformanceData"
"Library"="query.dll"
"Last Counter"=dword:00000840
"Last Help"=dword:00000841
"First Counter"=dword:0000083a
"First Help"=dword:0000083b
"Object List"="2106"
"WbemAdapFileTime"=hex:00,f7,5c,bd,a0,a5,bf,01
"WbemAdapFileSize"=dword:00158910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentIndex]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentIndex\Linkage]
"Bind"="\\Dummy"
"Export"="\\Dummy"
"Route"="\\Dummy"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ContentIndex\Performance]
"Close"="DoneCIPerformanceData"
"Collect"="CollectCIPerformanceData"
"Open"="InitializeCIPerformanceData"
"Library"="query.dll"
"Last Counter"=dword:00000838
"Last Help"=dword:00000839
"First Counter"=dword:00000822
"First Help"=dword:00000823
"Object List"="2082"
"WbemAdapFileTime"=hex:00,f7,5c,bd,a0,a5,bf,01
"WbemAdapFileSize"=dword:00158910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cpqarray]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000100
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cpqarray\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Cpqarray\Parameters\PnpInterfa
ce]
"2"=dword:00000001
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqarry2]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000000
"Tag"=dword:0000003e
"Type"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,70,71,\
  61,72,72,79,32,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqarry2\Parameters]
"BusType"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqarry2\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqarry2\Enum]
"0"="PCI\\VEN_1000&DEV_0010&SUBSYS_40400E11&REV_02\\3&267a616a&0&20"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,70,71,\
  63,69,73,73,62,2e,73,79,73,00
"DisplayName"="Compaq CISS Controllers Device Driver"
"Group"="port"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Parameters]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Enum]
"0"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&267a616a&0&38"
"Count"=dword:00000003
"NextInstance"=dword:00000003
"1"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&13c0b0c5&0&40"
"2"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&1070020&0&38"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,70,71,\
  63,69,73,73,64,2e,73,79,73,00
"DisplayName"="Compaq CISS Controllers Disk Driver"
"Group"="Primary Disk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,50,00,5f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,31,00,33,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,31,00,33,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Enum]
"0"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0000004000000000"
"Count"=dword:00000009
"NextInstance"=dword:00000009
"1"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0100004000000000"
"2"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0200004000000000"
"3"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0000004000000000"
"4"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0100004000000000"
"5"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0200004000000000"
"6"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0000004000000000"
"7"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0100004000000000"
"8"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0200004000000000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfcalm]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000002c
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfcalm\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfcalm\Parameters\PnpInterfa
ce]
"2"=dword:00000001
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfws2e]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000026
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfws2e\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqfws2e\Parameters\PnpInterfa
ce]
"2"=dword:00000001
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dac960nt]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000020
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dac960nt\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dac960nt\Parameters\PnpInterfa
ce]
"2"=dword:00000001
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\deckzpsx]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000002e
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\deckzpsx\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\deckzpsx\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dfs]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,44,66,73,73,76,63,2e,65,78,65,00
"DisplayName"="Distributed File System"
"Group"="Dfs"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  4c,61,6e,6d,61,6e,53,65,72,76,65,72,00,44,66,73,44,72,69,76,65,72,00,4d,75,\
  70,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Manages logical volumes distributed across a local or wide area
network."
"DfsVersion"=dword:00000565

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dfs\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dfs\Enum]
"0"="Root\\LEGACY_DFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DfsDriver]
"Type"=dword:00000002
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):73,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,44,66,73,\
  2e,73,79,73,00
"DisplayName"="DfsDriver"
"Group"="filter"
"Description"="Manages logical volumes distributed across a local or wide area
network."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DfsDriver\LocalVolumes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DfsDriver\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,42,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,49,00,4f,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,49,00,4f,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DfsDriver\Enum]
"0"="Root\\LEGACY_DFSDRIVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="DHCP Client"
"Group"="TDI"
"DependOnService"=hex(7):54,63,70,69,70,00,41,66,64,00,4e,65,74,42,54,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Manages network configuration by registering and updating IP
addresses and DNS names."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Linkage]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Linkage\Disabled]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\1]
"KeyType"=dword:00000007
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,69,70,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,3f,5c,44,68,63,70,53,\
  75,62,6e,65,74,4d,61,73,6b,4f,70,74,00,53,59,53,54,45,4d,5c,43,75,72,72,65,\
  6e,74,43,6f,6e,74,72,6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,3f,5c,50,\
  61,72,61,6d,65,74,65,72,73,5c,54,63,70,69,70,5c,44,68,63,70,53,75,62,6e,65,\
  74,4d,61,73,6b,4f,70,74,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\15]
"KeyType"=dword:00000001
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,69,70,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,3f,5c,44,68,63,70,44,\
  6f,6d,61,69,6e,00,53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,49,70,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,44,68,63,70,44,6f,6d,61,69,6e,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\3]
"KeyType"=dword:00000007
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,69,70,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,3f,5c,44,68,63,70,44,\
  65,66,61,75,6c,74,47,61,74,65,77,61,79,00,53,59,53,54,45,4d,5c,43,75,72,72,\
  65,6e,74,43,6f,6e,74,72,6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,3f,5c,\
  50,61,72,61,6d,65,74,65,72,73,5c,54,63,70,69,70,5c,44,68,63,70,44,65,66,61,\
  75,6c,74,47,61,74,65,77,61,79,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\44]
"KeyType"=dword:00000001
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,4e,65,74,42,54,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,54,63,70,69,70,5f,3f,\
  5c,44,68,63,70,4e,61,6d,65,53,65,72,76,65,72,4c,69,73,74,00,53,59,53,54,45,\
  4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,6f,6c,53,65,74,5c,53,65,72,76,69,\
  63,65,73,5c,4e,65,74,42,54,5c,41,64,61,70,74,65,72,73,5c,3f,5c,44,68,63,70,\
  4e,61,6d,65,53,65,72,76,65,72,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\46]
"KeyType"=dword:00000004
"RegLocation"="SYSTEM\\CurrentControlSet\\Services\\NetBT\\Parameters\\DhcpNodeType"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\47]
"KeyType"=dword:00000001
"RegLocation"="SYSTEM\\CurrentControlSet\\Services\\NetBT\\Parameters\\DhcpScopeID"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\6]
"KeyType"=dword:00000001
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,69,70,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,3f,5c,44,68,63,70,4e,\
  61,6d,65,53,65,72,76,65,72,00,53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,\
  6f,6e,74,72,6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,54,63,70,69,70,5c,\
  50,61,72,61,6d,65,74,65,72,73,5c,44,68,63,70,4e,61,6d,65,53,65,72,76,65,72,\
  00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\Options\DhcpNe
tbiosOptions]
"KeyType"=dword:00000004
"OptionId"=dword:00000001
"VendorType"=dword:00000001
"RegLocation"=hex(7):53,59,53,54,45,4d,5c,43,75,72,72,65,6e,74,43,6f,6e,74,72,\
  6f,6c,53,65,74,5c,53,65,72,76,69,63,65,73,5c,4e,65,74,42,54,5c,50,61,72,61,\
  6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,54,63,70,69,70,5f,3f,\
  5c,44,68,63,70,4e,65,74,62,69,6f,73,4f,70,74,69,6f,6e,73,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
```

```
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dhcp\Enum]
"0"="Root\\LEGACY_DHCP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Disk]
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,6f,72,74,00,00
"ErrorControl"=dword:00000001
"Group"="SCSI Class"
"Start"=dword:00000000
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="Disk Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,64,69,73,\
  6b,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Disk\Enum]
"0"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&000"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&010"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Diskperf]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000000
"Tag"=dword:0000000a
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Diskperf\Enum]
"0"="Root\\LEGACY_DISKPERF\\0000"
"Count"=dword:0000000c
"NextInstance"=dword:0000000c
"1"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&000"
"2"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&010"
"3"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0000004000000000"
"4"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0100004000000000"
"5"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0200004000000000"
"6"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0000004000000000"
"7"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0100004000000000"
"8"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0200004000000000"
"9"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0000004000000000"
"10"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0100004000000000"
"11"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0200004000000000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmadmin]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,64,6d,61,64,6d,69,6e,2e,65,78,65,20,2f,63,6f,6d,00
"DisplayName"="Logical Disk Manager Administrative Service"
"ObjectName"="LocalSystem"
"Description"="Administrative service for disk management requests"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmadmin\Enum]
"0"="Root\\LEGACY_DMADMIN\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmboot]
"Type"=dword:00000001
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"Group"="Filter"
"Tag"=dword:0000000b
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,64,6d,62,\
  6f,6f,74,2e,73,79,73,00
"VolumeRecoveryNeeded"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmboot\Enum]
"0"="Root\\LEGACY_DMBOOT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmio]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Tag"=dword:0000000d
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,64,6d,69,\
  6f,2e,73,79,73,00
"DisplayName"="Logical Disk Manager Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmio\Boot Info]
"Boot ID"="92798d41-6c75-11d4-bdfb-806d6172696f"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmio\Enum]
"0"="Root\\dmio\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmload]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Tag"=dword:0000000c
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,64,6d,6c,\
  6f,61,64,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmload\Enum]
"0"="Root\\LEGACY_DMLOAD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmserver]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Logical Disk Manager"
"ObjectName"="LocalSystem"
"Description"="Logical Disk Manager Watchdog Service"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmserver\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
```

```
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\dmserver\Enum]
"0"="Root\\LEGACY_DMSERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="DNS Client"
"Group"="TDI"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Resolves and caches Domain Name System (DNS) names."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Parameters]
"NegativeCacheTime"=dword:0000012c
"NegativeSOACacheTime"=dword:00000078
"NetFailureCacheTime"=dword:00000000
"AdapterTimeoutCacheTime"=dword:00000000
"CacheHashTableSize"=dword:000000d3
"CacheHashTableBucketSize"=dword:0000000a
"NetFailureErrorPopupLimit"=dword:00000000
"MaxCacheEntryTtlLimit"=dword:00015180
"MaxSOACacheEntryTtlLimit"=dword:0000012c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Dnscache\Enum]
"0"="Root\\LEGACY_DNSCACHE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EFS]
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000004
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EFS\Enum]
"0"="Root\\LEGACY_EFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog]
"Description"="Logs event messages issued by programs and Windows.  Event Log
reports contain information that can be useful in diagnosing problems.  Reports are
viewed in Event Viewer."
"DisplayName"="Event Log"
"ErrorControl"=dword:00000001
"Group"="Event log"
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"PlugPlayServiceType"=dword:00000003
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application]
"DisplayNameFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,65,6c,73,2e,64,6c,6c,00
"DisplayNameID"=dword:00000100
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,63,6f,6e,66,69,67,5c,41,70,70,45,76,65,6e,74,2e,45,76,74,00
"MaxSize"=dword:00080000
"PrimaryModule"="Application"
"Retention"=dword:00093a80
@="mnmsrvc"
"Sources"=hex(7):57,53,48,00,57,69,6e,4d,67,6d,74,00,57,69,6e,6c,6f,67,6f,6e,\
  00,57,69,6e,64,6f,77,73,20,33,2e,31,20,4d,69,67,72,61,74,69,6f,6e,00,56,42,\
  52,75,6e,74,69,6d,65,00,55,73,65,72,69,6e,69,74,00,55,73,65,72,65,6e,76,00,\
  54,6c,6e,74,73,76,72,00,53,79,73,6d,6f,6e,4c,6f,67,00,53,51,4c,53,45,52,56,\
  45,52,41,47,45,4e,54,00,53,51,4c,43,54,52,00,53,70,6f,6f,6c,65,72,43,74,72,\
  73,00,53,6f,66,74,77,61,72,65,20,49,6e,73,74,61,6c,6c,61,74,69,6f,6e,00,53,\
  63,6c,67,4e,74,66,79,00,53,63,65,53,72,76,00,53,63,65,43,6c,69,00,50,6c,75,\
  67,50,6c,61,79,4d,61,6e,61,67,65,72,00,50,65,72,66,50,72,6f,63,00,50,65,72,\
  66,4f,53,00,50,65,72,66,4e,65,74,00,50,65,72,66,6d,6f,6e,00,50,65,72,66,6c,\
  69,62,00,50,65,72,66,44,69,73,6b,00,50,65,72,66,63,74,72,73,00,4f,66,66,6c,\
  69,6e,65,20,46,69,6c,65,73,00,4f,61,6b,6c,65,79,00,6e,74,62,61,63,6b,75,70,\
  00,4d,53,53,51,4c,53,65,72,76,65,72,41,44,48,65,6c,70,65,72,00,4d,53,53,51,\
  4c,53,45,52,56,45,52,00,4d,73,69,49,6e,73,74,61,6c,6c,65,72,00,4d,53,44,54,\
  43,20,43,6c,69,65,6e,74,00,4d,53,44,54,43,00,6d,6e,6d,73,72,76,63,00,4c,6f,\
  61,64,50,65,72,66,00,4c,69,63,65,6e,73,65,53,65,72,76,69,63,65,00,4a,61,76,\
  61,20,56,4d,00,49,50,53,45,43,50,6f,6c,69,63,79,53,74,6f,72,61,67,65,00,68,\
  70,6d,6f,6e,00,46,6f,6c,64,65,72,20,52,65,64,69,72,65,63,74,69,6f,6e,00,46,\
  69,6c,65,20,44,65,70,6c,6f,79,6d,65,6e,74,00,45,76,65,6e,74,53,79,73,74,65,\
  6d,00,45,53,45,4e,54,00,44,72,57,61,74,73,6f,6e,00,44,69,73,6b,51,75,6f,74,\
  61,00,44,61,74,61,54,72,61,6e,73,66,6f,72,6d,61,74,69,6f,6e,53,65,72,76,69,\
  63,65,73,00,43,4f,4d,2b,00,43,69,00,43,68,6b,64,73,6b,00,41,75,74,6f,63,68,\
  6b,00,41,70,70,6c,69,63,61,74,69,6f,6e,20,4d,61,6e,61,67,65,6d,65,6e,74,00,\
  41,70,70,6c,69,63,61,74,69,6f,6e,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Applicati
on]
"CategoryCount"=dword:00000007
"CategoryMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,\
  74,65,6d,33,32,5c,65,76,65,6e,74,6c,6f,67,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Applicati
on Management]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,61,70,70,6d,67,6d,74,73,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Autochk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,69,6e,6c,6f,67,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Chkdsk]
```

```
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,75,6c,69,62,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Ci]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,71,75,65,72,79,2e,64,6c,6c,00
"CategoryMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,71,75,65,72,79,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
"CategoryCount"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\COM+]
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,\
  5c,43,4f,4d,53,56,43,53,2e,44,4c,4c,00
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,\
  32,5c,43,4f,4d,53,56,43,53,2e,44,4c,4c,00
"ParameterMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,\
  32,5c,43,4f,4d,53,56,43,53,2e,44,4c,4c,00
"TypeSupported"=dword:00000007
"CategoryCount"=dword:00000014

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\DataTrans
formationServices]
"EventMessageFile"="C:\\Program Files\\Microsoft SQL
Server\\80\\Tools\\Binn\\Resources\\1033\\dtspkg.RLL"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\DiskQuota
]
"EventMessageFile"="%SystemRoot%\\System32\\dskquota.dll"
"TypesSupported"="0x00000007"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\DrWatson]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,64,72,77,74,73,6e,33,32,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\ESENT]
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,73,79,73,74,65,6d,33,32,\
  5c,45,53,45,4e,54,2e,64,6c,6c,00
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,73,79,73,74,65,6d,33,\
  32,5c,45,53,45,4e,54,2e,64,6c,6c,00
"CategoryCount"=dword:0000000f
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\EventSyst
em]
"CategoryCount"=dword:00000006
"TypesSupported"=dword:00000007
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,\
  32,5c,65,73,2e,64,6c,6c,00
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,\
  5c,65,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\File
Deployment]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,66,64,65,70,6c,6f,79,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Folder
Redirection]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,66,64,65,70,6c,6f,79,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\hpmon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,68,70,6d,6f,6e,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\IPSECPoli
cyStorage]
"EventMessageFile"="%SystemRoot%\\System32\\polstore.dll"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Java VM]
"EventMessageFile"=hex(2):25,73,79,73,74,65,6d,72,6f,6f,74,25,5c,73,79,73,74,\
  65,6d,33,32,5c,76,6d,68,65,6c,70,65,72,2e,64,6c,6c,00
"TypesSupported"=hex:07,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\LicenseSe
rvice]
"TypesSupported"=dword:00000007
"EventMessageFile"="%SystemRoot%\\System32\\llsrpc.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\LoadPerf]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6c,6f,61,64,70,65,72,66,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\mnmsrvc]
"EventMessageFile"="%SystemRoot%\\System32\\nmevtmsg.dll"
"TypeSupported"=hex:07,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MSDTC]
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,\
  5c,4d,53,44,54,43,50,52,58,2e,44,4c,4c,00
"TypesSupported"=dword:00000007
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,\
  32,5c,4d,53,44,54,43,50,52,58,2e,44,4c,4c,00
"CategoryCount"=dword:00000012

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MSDTC
Client]
"EventMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,\
  5c,4d,53,44,54,43,50,52,58,2e,44,4c,4c,00
"TypesSupported"=dword:00000007
"CategoryMessageFile"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,\
  32,5c,4d,53,44,54,43,50,52,58,2e,44,4c,4c,00
"CategoryCount"=dword:00000012

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MsiInstal
ler]
"EventMessageFile"="C:\\WINNT\\system32\\Msi.dll"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MSSQLSERV
ER]
"CategoryCount"=dword:00000008
"TypesSupported"=dword:000000ff
```

```
"EventMessageFile"="C:\\PROGRA~1\\MICROS~2\\MSSQL\\BINN\\RESOUR~1\\1033\\SQLEVN70.RL
L"
"CategoryMessageFile"="C:\\PROGRA~1\\MICROS~2\\MSSQL\\BINN\\RESOUR~1\\1033\\SQLEVN70
.RLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\MSSQLServ
erADHelper]
"EventMessageFile"=hex(2):43,3a,5c,50,72,6f,67,72,61,6d,20,46,69,6c,65,73,5c,\
  4d,69,63,72,6f,73,6f,66,74,20,53,51,4c,20,53,65,72,76,65,72,5c,38,30,5c,54,\
  6f,6f,6c,73,5c,42,69,6e,6e,5c,52,65,73,6f,75,72,63,65,73,5c,31,30,33,33,5c,\
  73,71,6c,61,64,65,76,6e,2e,72,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\ntbackup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,74,62,61,63,6b,75,70,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Oakley]
"EventMessageFile"="%SystemRoot%\\System32\\oakley.dll"
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Offline
Files]
"EventMessageFile"="%SystemRoot%\\System32\\cscui.dll"
"TypesSupported"="0x00000007"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Perfctrs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,63,74,72,73,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\PerfDisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,64,69,73,6b,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Perflib]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,72,66,6c,62,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Perfmon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,6d,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\PerfNet]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,6e,65,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\PerfOS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,4f,53,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\PerfProc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,65,72,66,70,72,6f,63,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\PlugPlayM
anager]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,75,6d,70,6e,70,6d,67,72,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SceCli]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,63,65,63,6c,69,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SceSrv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,63,65,73,72,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SclgNtfy]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,63,6c,67,6e,74,66,79,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Software
Installation]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,61,70,70,6d,67,72,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SpoolerCt
rs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,69,6e,73,70,6f,6f,6c,2e,64,72,76,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SQLCTR]
"TypesSupported"=dword:00000007
"EventMessageFile"="C:\\PROGRA~1\\MICROS~2\\MSSQL\\BINN\\RESOUR~1\\1033\\SQLEVN70.RL
L"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SQLSERVER
AGENT]
"CategoryCount"=dword:00000004
"TypesSupported"=dword:00000007
"EventMessageFile"="C:\\PROGRA~1\\MICROS~2\\MSSQL\\BINN\\SQLAGENT.DLL"
"CategoryMessageFile"="C:\\PROGRA~1\\MICROS~2\\MSSQL\\BINN\\SQLAGENT.DLL"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\SysmonLog
]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,6d,6c,6f,67,73,76,63,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Tlntsvr]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,74,6c,6e,74,73,76,72,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Userenv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,75,73,65,72,65,6e,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Userinit]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
```

```
  65,6d,33,32,5c,75,73,65,72,69,6e,69,74,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\VBRuntime
]
"EventMessageFile"="C:\\WINNT\\System32\\msvbvm60.dll"
"TypesSupported"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Windows
3.1 Migration]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,61,64,76,61,70,69,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\Winlogon]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,69,6e,6c,6f,67,6f,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\WinMgmt]
"EventMessageFile"="C:\\WINNT\\System32\\WBEM\\WinMgmtR.dll"
"TypesSupported"=hex:07

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Application\WSH]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,73,68,65,78,74,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security]
"DisplayNameFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,65,6c,73,2e,64,6c,6c,00
"DisplayNameID"=dword:00000101
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,63,6f,6e,66,69,67,5c,53,65,63,45,76,65,6e,74,2e,45,76,74,00
"MaxSize"=dword:00080000
"PrimaryModule"="Security"
"Retention"=dword:00093a80
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"Sources"=hex(7):53,70,6f,6f,6c,65,72,00,53,65,63,75,72,69,74,79,20,41,63,63,\
  6f,75,6e,74,20,4d,61,6e,61,67,65,72,00,53,43,20,4d,61,6e,61,67,65,72,00,4e,\
  65,74,44,44,45,20,4f,62,6a,65,63,74,00,4c,53,41,00,44,53,00,53,65,63,75,72,\
  69,74,79,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\DS]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\DS\ObjectNam
es]
"Directory Service Object"=dword:00001e00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\LSA]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\LSA\ObjectNa
mes]
"PolicyObject"=dword:00001600
"SecretObject"=dword:00001610
"TrustedDomainObject"=dword:00001620
"UserAccountObject"=dword:00001630

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\NetDDE
Object]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\NetDDE
Object\ObjectNames]
"DDE Share"=dword:00001d00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\SC Manager]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\SC
Manager\ObjectNames]
"SC_MANAGER Object"=dword:00001c00
"SERVICE Object"=dword:00001c10

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Security]
"CategoryCount"=dword:00000009
"CategoryMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,41,75,64,69,74,45,2e,64,6c,6c,00
"GuidMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,\
  6d,33,32,5c,4e,74,4d,61,72,74,61,2e,64,6c,6c,00
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4d,73,41,75,64,69,74,45,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00
"TypesSupported"=dword:0000001c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Security\Obj
ectNames]
"Channel"=dword:00001400
"Desktop"=dword:00001a10
"Device"=dword:00001100
"Directory"=dword:00001110
"Event"=dword:00001120
"EventPair"=dword:00001130
"File"=dword:00001140
"IoCompletion"=dword:00001300
"Job"=dword:00001410
"Key"=dword:00001150
"MailSlot"=dword:00001140
"Mutant"=dword:00001160
"NamedPipe"=dword:00001140
"Port"=dword:00001170
"Process"=dword:00001180
"Profile"=dword:00001190
"Section"=dword:000011a0
"Semaphore"=dword:000011b0
"SymbolicLink"=dword:000011c0
"Thread"=dword:000011d0
"Timer"=dword:000011e0
"Token"=dword:000011f0
"Type"=dword:00001200
"WaitablePort"=dword:00001170
```

```
"WindowStation"=dword:00001a00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Security
Account Manager]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Security
Account Manager\ObjectNames]
"SAM_ALIAS"=dword:00001530
"SAM_DOMAIN"=dword:00001510
"SAM_GROUP"=dword:00001520
"SAM_SERVER"=dword:00001500
"SAM_USER"=dword:00001540

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Spooler]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,4d,73,4f,62,6a,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security\Spooler\Objе
ctNames]
"Document"=dword:00001b20
"Printer"=dword:00001b10
"Server"=dword:00001b00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System]
"DisplayNameFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,65,6c,73,2e,64,6c,6c,00
"DisplayNameID"=dword:00000102
"File"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,63,6f,6e,66,69,67,5c,53,79,73,45,76,65,6e,74,2e,45,76,74,00
"MaxSize"=dword:00080000
"PrimaryModule"="System"
"Retention"=dword:00093a80
"Sources"=hex(7):57,6f,72,6b,73,74,61,74,69,6f,6e,00,57,6d,69,00,57,69,6e,64,\
  6f,77,73,20,46,69,6c,65,20,50,72,6f,74,65,63,74,69,6f,6e,00,57,69,6e,33,32,\
  6b,00,57,33,32,54,69,6d,65,00,56,67,61,53,61,76,65,00,55,50,53,00,75,6c,74,\
  72,61,36,36,00,75,64,66,73,00,54,65,72,6d,53,65,72,76,69,63,65,00,54,65,72,\
  6d,53,65,72,76,44,65,76,69,63,65,73,00,54,65,72,6d,44,44,00,74,64,69,00,54,\
  43,50,4d,6f,6e,00,54,63,70,69,70,00,73,79,6d,5f,68,69,00,73,79,6d,63,38,78,\
  78,00,73,79,6d,63,38,31,30,00,53,74,69,6c,6c,49,6d,61,67,65,00,53,72,76,00,\
  73,70,61,72,72,6f,77,00,73,6e,64,62,6c,73,74,00,53,69,6d,62,61,64,00,73,67,\
  6c,66,62,00,73,66,6c,6f,70,70,79,00,53,65,72,76,69,63,65,20,43,6f,6e,74,72,\
  6f,6c,20,4d,61,6e,61,67,65,72,00,53,65,72,76,65,72,00,73,65,72,69,61,6c,00,\
  73,63,73,69,70,6f,72,74,00,53,63,68,65,64,75,6c,65,00,53,43,61,72,64,53,76,\
  72,00,53,61,76,65,20,44,75,6d,70,00,53,41,4d,00,52,53,56,50,00,52,65,6d,6f,\
  76,61,62,6c,65,20,53,74,6f,72,61,67,65,20,53,65,72,76,69,63,65,00,52,65,6d,\
  6f,74,65,41,63,63,65,73,73,00,72,65,64,62,6f,6f,6b,00,52,64,62,73,73,00,52,\
  61,73,4d,61,6e,00,52,61,73,41,75,74,6f,00,71,6c,32,31,30,30,00,71,6c,31,32,\
  34,30,00,71,6c,31,30,77,6e,74,00,71,6c,31,30,38,30,00,50,72,69,6e,74,00,50,\
  70,74,70,4d,69,6e,69,70,6f,72,74,00,50,6f,6c,69,63,79,41,67,65,6e,74,00,70,\
  63,6d,63,69,61,00,70,63,69,69,64,65,00,70,63,69,00,70,61,72,76,64,6d,00,70,\
  61,72,70,6f,72,74,00,70,61,72,61,6c,6c,65,6c,00,4f,53,50,46,4d,69,62,00,4f,\
  53,50,46,00,6e,75,6c,6c,00,4e,74,53,65,72,76,69,63,65,50,61,63,6b,00,4e,54,\
  4d,53,00,6e,74,66,73,00,6e,70,66,73,00,4e,65,74,6c,6f,67,6f,6e,00,4e,65,74,\
  44,44,45,00,4e,65,74,42,54,00,4e,65,74,42,49,4f,53,00,4e,64,69,73,57,61,6e,\
  00,6e,64,69,73,00,6e,63,72,63,37,31,30,00,4e,31,30,30,00,4d,75,70,00,6d,73,\
  66,73,00,6d,73,61,64,6c,69,62,00,4d,72,78,53,6d,62,00,6d,72,61,69,64,33,35,\
  78,00,6d,6f,75,63,6c,61,73,73,00,4d,6f,64,65,6d,00,4c,73,61,53,72,76,00,6c,\
  70,36,6e,64,73,33,35,00,4c,6d,48,6f,73,74,73,00,4c,44,4d,53,00,4c,44,4d,00,\
  6c,62,72,74,66,64,63,00,4b,65,72,62,65,72,6f,73,00,4b,44,43,00,6b,62,64,63,\
  6c,61,73,73,00,69,73,61,70,6e,70,00,49,50,58,53,41,50,00,49,50,58,52,6f,75,\
  74,65,72,4d,61,6e,61,67,65,72,00,49,50,58,52,49,50,00,49,50,58,43,50,00,69,\
  70,73,72,61,69,64,6e,00,49,50,53,45,43,00,49,50,52,6f,75,74,65,72,4d,61,6e,\
  61,67,65,72,00,49,50,52,49,50,32,00,49,50,4e,41,54,48,4c,50,00,49,50,42,4f,\
  4f,54,50,00,69,6e,74,65,6c,69,64,65,00,69,6e,69,39,31,30,75,00,69,38,30,34,\
  32,70,72,74,00,66,74,64,69,73,6b,00,66,73,5f,72,65,63,00,66,6c,70,79,64,69,\
  73,6b,00,66,6c,61,73,68,70,6e,74,00,66,69,72,65,70,6f,72,74,00,66,64,63,00,\
  66,64,31,36,5f,37,30,30,00,66,61,73,74,66,61,74,00,65,76,65,6e,74,6c,6f,67,\
  00,65,66,73,00,44,6e,73,63,61,63,68,65,00,44,6e,73,61,70,69,00,64,6d,69,6f,\
  00,64,6d,62,6f,6f,74,00,44,69,73,74,72,69,62,75,74,65,64,20,4c,69,6e,6b,20,\
  54,72,61,63,6b,69,6e,67,20,53,65,72,76,65,72,00,44,69,73,74,72,69,62,75,74,\
  65,64,20,4c,69,6e,6b,20,54,72,61,63,6b,69,6e,67,20,43,6c,69,65,6e,74,00,64,\
  69,73,6b,70,65,72,66,00,64,69,73,6b,00,44,68,63,70,00,44,66,73,53,76,63,00,\
  44,66,73,44,72,69,76,65,72,00,64,65,63,6b,7a,70,73,78,00,44,43,4f,4d,00,64,\
  61,63,39,36,30,6e,74,00,63,70,71,66,77,73,32,65,00,63,70,71,66,63,61,6c,6d,\
  00,63,70,71,63,69,73,73,64,00,63,70,71,63,69,73,73,62,00,63,70,71,61,72,72,\
  79,32,00,63,70,71,61,72,72,61,79,00,63,68,61,6e,67,65,72,00,63,64,72,6f,6d,\
  00,43,64,6d,00,63,64,66,73,00,63,64,61,75,64,69,6f,00,63,64,32,30,78,72,6e,\
  74,00,62,75,73,6c,6f,67,69,63,00,42,72,6f,77,73,65,72,00,62,65,65,70,00,41,\
  74,6d,61,72,70,63,00,61,74,69,72,61,67,65,00,61,74,64,69,73,6b,00,61,74,61,\
  70,69,00,41,73,79,6e,63,4d,61,63,00,61,73,63,33,35,35,30,00,61,73,63,33,33,\
  35,30,70,00,61,73,63,00,41,70,70,6c,69,63,61,74,69,6f,6e,20,50,6f,70,75,70,\
  00,61,6d,73,69,6e,74,00,61,6d,69,30,6e,74,00,41,6c,65,72,74,65,72,00,61,69,\
  63,37,38,78,78,00,61,69,63,37,38,75,32,00,61,69,63,31,31,36,78,00,61,68,61,\
  31,35,34,78,00,61,64,70,75,31,36,30,6d,00,61,63,70,69,65,63,00,61,63,70,69,\
  00,61,62,70,34,38,30,6e,35,00,61,62,69,6f,73,64,73,6b,00,53,79,73,74,65,6d,\
  00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\abiosdsk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\abp480n5]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\acpi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,61,63,\
  70,69,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\acpiec]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,61,63,\
  70,69,65,63,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\adpu160m]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\aha154x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\aic116x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
```

```
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\aic78u2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\aic78xx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Alerter]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ami0nt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\amsint]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Application
Popup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,74,64,6c,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\asc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\asc3350p]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\asc3550]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\AsyncMac]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\atapi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\atdisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\atirage]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,61,74,\
  69,72,61,67,65,6d,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Atmarpc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\beep]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Browser]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\buslogic]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cd20xrnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cdaudio]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cdfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Cdm]
@=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cdrom]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\changer]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqarray]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqarry2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
```

```
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqcissb]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,44,72,69,76,65,72,73,5c,43,50,51,43,49,53,53,42,2e,53,59,53,\
  00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqcissd]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,44,72,69,76,65,72,73,5c,43,50,51,43,49,53,53,44,2e,53,59,53,\
  00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqfcalm]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\cpqfws2e]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\dac960nt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\DCOM]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\deckzpsx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\DfsDriver]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\DfsSvc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Dhcp]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,64,68,63,70,63,73,76,63,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\disk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\diskperf]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Distributed
Link Tracking Client]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Distributed
Link Tracking Server]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\dmboot]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,44,72,69,76,65,72,73,5c,64,6d,62,6f,6f,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\dmio]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,64,6d,\
  69,6f,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Dnsapi]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Dnscache]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\efs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6c,73,61,73,72,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\eventlog]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\fastfat]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\fd16_700]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\fdc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
```

```
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,66,64,\
  63,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\fireport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\flashpnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\flpydisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,66,6c,\
  70,79,64,69,73,6b,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\fs_rec]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ftdisk]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,46,74,\
  44,69,73,6b,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\i8042prt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,69,38,\
  30,34,32,70,72,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ini910u]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\intelide]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,49,6e,\
  74,65,6c,49,64,65,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPBOOTP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,69,70,62,6f,6f,74,70,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPNATHLP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,69,70,6e,61,74,68,6c,70,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPRIP2]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,69,70,72,69,70,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPRouterManage
r]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPSEC]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ipsraidn]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPXCP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPXRIP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPXRouterManag
er]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\IPXSAP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\isapnp]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,69,73,\
  61,70,6e,70,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\kbdclass]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,6b,62,\
  64,63,6c,61,73,73,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\KDC]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6b,64,63,73,76,63,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Kerberos]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6b,65,72,62,65,72,6f,73,2e,64,6c,6c,00
```

```
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\lbrtfdc]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,6c,62,\
  72,74,66,64,63,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\LDM]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,64,6d,61,64,6d,69,6e,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\LDMS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,64,6d,73,65,72,76,65,72,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\LmHosts]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\lp6nds35]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\LsaSrv]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6c,73,61,73,72,76,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Modem]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,4d,6f,\
  64,65,6d,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\mouclass]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,6d,6f,\
  75,63,6c,61,73,73,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\mraid35x]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\MrxSmb]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,69,6f,6c,6f,67,6d,73,67,2e,64,\
  6c,6c,00
"TypesSupported"=dword:00000007
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00,5c

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\msadlib]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\msfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Mup]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\N100]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,6e,31,\
  30,30,6e,74,35,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ncrc710]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ndis]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NdisWan]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NetBIOS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,69,6f,6c,6f,67,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NetBT]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NetDDE]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Netlogon]
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\npfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ntfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NTMS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,\
  65,6d,33,32,5c,4e,74,6d,73,45,76,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\NtServicePack]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,70,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\null]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\OSPF]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6f,73,70,66,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\OSPFMib]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6f,73,70,66,6d,69,62,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\parallel]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,70,61,\
  72,61,6c,6c,65,6c,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\parport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,70,61,\
  72,70,6f,72,74,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\parvdm]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,50,61,\
  72,56,64,6d,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\pci]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,50,63,\
  69,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\pciide]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,50,63,\
  69,49,64,65,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\pcmcia]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,50,63,\
  6d,63,69,61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\PolicyAgent]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,70,6f,6c,61,67,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\PptpMiniport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Print]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4c,6f,63,61,6c,53,70,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ql1080]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ql10wnt]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ql1240]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ql2100]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\RasAuto]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\RasMan]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Rdbss]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\redbook]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,72,65,\
```

```
  64,62,6f,6f,6b,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\RemoteAccess]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6d,70,72,6d,73,67,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,69,61,73,73,76,63,73,2e,64,6c,6c,00
"TypesSupported"=dword:0000001f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Removable
Storage Service]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4e,54,4d,53,45,56,54,2e,44,4c,4c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\RSVP]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,72,73,76,70,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\SAM]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,61,6d,73,72,76,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Save Dump]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,53,61,76,65,44,75,6d,70,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\SCardSvr]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,53,43,61,72,64,53,76,72,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Schedule]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\scsiport]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\serial]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,73,65,\
  72,69,61,6c,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Server]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Service
Control Manager]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"ParameterMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,\
  74,65,6d,33,32,5c,6b,65,72,6e,65,6c,33,32,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\sfloppy]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\sglfb]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,73,67,\
  6c,66,62,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Simbad]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\sndblst]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\sparrow]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Srv]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\StillImage]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,74,69,73,76,63,2e,65,78,65,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\symc810]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\symc8xx]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\sym_hi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\System]
"CategoryCount"=dword:00000007
"CategoryMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,\
  74,65,6d,33,32,5c,65,76,65,6e,74,6c,6f,67,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Tcpip]
```

```
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\TCPMon]
"TypesSupported"=dword:00000007
"EventMessageFile"="%SystemRoot%\\System32\\tcpmon.dll"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\tdi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\TermDD]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,74,64,6c,6c,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\TermServDevice
s]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,6c,6e,6f,74,69,66,79,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\TermService]
"TypesSupported"=dword:00000007
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,74,65,72,6d,73,72,76,2e,65,78,65,3b,25,53,79,73,74,65,6d,52,\
  6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,6e,74,64,6c,6c,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\udfs]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\ultra66]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\UPS]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\VgaSave]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,49,6f,4c,6f,67,4d,73,67,2e,64,6c,6c,3b,25,53,79,73,74,65,6d,\
  52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,5c,44,72,69,76,65,72,73,5c,76,67,\
  61,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\W32Time]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,33,32,74,69,6d,65,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Win32k]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,77,69,6e,33,32,6b,2e,73,79,73,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Windows File
Protection]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,73,66,63,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Wmi]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,65,76,65,6e,74,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\System\Workstation]
"EventMessageFile"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,6e,65,74,6d,73,67,2e,64,6c,6c,00
"TypesSupported"=dword:00000007

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventSystem]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,73,76,\
  63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"DisplayName"="COM+ Event System"
"Group"="Network"
"DependOnService"=hex(7):52,50,43,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides automatic distribution of events to subscribing COM
components."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventSystem\Parameters]
"ServiceDll"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,65,\
  73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventSystem\Security]
"Security"=hex:01,00,14,80,b4,00,00,00,c0,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,84,00,05,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,72,00,73,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,72,00,73,00,00,00,14,00,9d,01,02,00,01,01,00,\
  00,00,00,00,01,00,00,00,00,00,01,01,00,00,00,00,00,00,05,12,00,00,00,01,01,00,00,\
  00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventSystem\Enum]
"0"="Root\\LEGACY_EVENTSYSTEM\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fastfat]
"ErrorControl"=dword:00000001
"Group"="Boot file system"
"Start"=dword:00000004
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fastfat\Enum]
"0"="Root\\LEGACY_FASTFAT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fax]
"Type"=dword:00000110
```

```
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,73,79,73,74,65,6d,72,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,66,61,78,73,76,63,2e,65,78,65,00
"DisplayName"="Fax Service"
"DependOnService"=hex(7):54,61,70,69,53,72,76,00,52,70,63,53,73,00,50,6c,75,67,\
  50,6c,61,79,00,53,70,6f,6f,6c,65,72,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Helps you send and receive faxes"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fax\Performance]
"Open"=hex(2):4f,70,65,6e,46,61,78,50,65,72,66,6f,72,6d,61,6e,63,65,44,61,74,\
  61,00
"Close"=hex(2):43,6c,6f,73,65,46,61,78,50,65,72,66,6f,72,6d,61,6e,63,65,44,61,\
  74,61,00
"Collect"=hex(2):43,6f,6c,6c,65,63,74,46,61,78,50,65,72,66,6f,72,6d,61,6e,63,\
  65,44,61,74,61,00
"Library"=hex(2):25,73,79,73,74,65,6d,72,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,66,61,78,70,65,72,66,2e,64,6c,6c,00
"Last Counter"=dword:00000820
"Last Help"=dword:00000821
"First Counter"=dword:00000802
"First Help"=dword:00000803
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00001910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fax\Security]
"Security"=hex:01,00,14,80,a0,01,00,00,ac,01,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,01,05,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\
  00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,\
  14,00,14,00,00,00,01,01,00,00,00,00,00,01,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,01,01,00,00,00,\
  00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fd16_700]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000010
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fdc]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000003
"Tag"=dword:00000002
"Type"=dword:00000001
"SetupDone"=dword:00000001
"DisplayName"="Floppy Disk Controller Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,66,64,63,\
  2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fdc\Enum]
"0"="ACPI\\PNP0700\\5&24f21e77&0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fireport]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000002d
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fireport\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\fireport\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\flashpnt]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000025
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\flashpnt\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\flashpnt\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Flpydisk]
"ErrorControl"=dword:00000001
"Group"="Primary disk"
"Start"=dword:00000003
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="Floppy Disk Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,66,6c,70,\
  79,64,69,73,6b,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Flpydisk\Enum]
"0"="FDC\\GENERIC_FLOPPY_DRIVE\\6&2236c331&0&0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fs_Rec]
"ErrorControl"=dword:00000000
"Group"="Boot file system"
"Start"=dword:00000001
"Type"=dword:00000008

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Fs_Rec\Enum]
"0"="Root\\LEGACY_FS_REC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ftdisk]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
```

```
"Start"=dword:00000000
"Tag"=dword:00000009
"Type"=dword:00000001
"DisplayName"="Volume Manager Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,66,74,64,\
  69,73,6b,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ftdisk\Enum]
"0"="Root\\ftdisk\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Gpc]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000003
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6d,73,67,\
  70,63,2e,73,79,73,00
"DisplayName"="Generic Packet Classifier"
"Group"="PNP_TDI"
"Description"="Generic Packet Classifier"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Gpc\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Gpc\Enum]
"0"="Root\\LEGACY_GPC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\i8042prt]
"Type"=dword:00000001
"Start"=dword:00000001
"Group"="Keyboard Port"
"ErrorControl"=dword:00000001
"DisplayName"="i8042 Keyboard and PS/2 Mouse Port Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,38,30,\
  34,32,70,72,74,2e,73,79,73,00
"Tag"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\i8042prt\Parameters]
"LayerDriver JPN"="kbd101.dll"
"LayerDriver KOR"="kbd101a.dll"
"PollingIterations"=dword:00002ee0
"PollingIterationsMaximum"=dword:00002ee0
"ResendIterations"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\i8042prt\Enum]
"0"="ACPI\\PNP0F13\\4&f0b8f99&0"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="ACPI\\PNP0303\\4&f0b8f99&0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IAS]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IAS\Performance]
"Library"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,69,61,73,70,65,72,66,2e,64,6c,6c,00
"Open"="OpenPerformanceData"
"Close"="ClosePerformanceData"
"Collect"="CollectPerformanceData"
"Last Counter"=dword:00000924
"Last Help"=dword:00000925
"First Counter"=dword:000008da
"First Help"=dword:000008db
"Object List"="2266 2268 2270 2272"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00005110
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\inetaccs]
@=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\inetaccs\Parameters]
@=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ini910u]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000030
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ini910u\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ini910u\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Inport]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Inport\Parameters]
"HzMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IntelIde]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000004
"Tag"=dword:00000004
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpFilterDriver]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,70,66,\
  6c,74,64,72,76,2e,73,79,73,00
"DisplayName"="IP Traffic Filter Driver"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"Description"="IP Traffic Filter Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpFilterDriver\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,fe,2e,d7,29,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
```

```
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,fe,2e,d7,29,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpInIp]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,70,69,\
  6e,69,70,2e,73,79,73,00
"DisplayName"="IP in IP Tunnel Driver"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"Description"="IP in IP Tunnel Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpInIp\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpNat]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,70,6e,\
  61,74,2e,73,79,73,00
"DisplayName"="IP Network Address Translator"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"Description"="IP Network Address Translator"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IpNat\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000003
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,70,73,\
  65,63,2e,73,79,73,00
"DisplayName"="IPSEC driver"
"Group"="Extended base"
"Description"="IPSEC driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,70,00,63,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IPSEC\Enum]
"0"="Root\\LEGACY_IPSEC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ipsraidn]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000035
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ipsraidn\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ipsraidn\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ISAPISearch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ISAPISearch\Linkage]
"Bind"="\\Dummy"
"Export"="\\Dummy"
"Route"="\\Dummy"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ISAPISearch\Performance]
"Close"="DoneCIISAPIPerformanceData"
"Collect"="CollectCIISAPIPerformanceData"
"Open"="InitializeCIISAPIPerformanceData"
"Library"="query.dll"
"Last Counter"=dword:00000856
"Last Help"=dword:00000857
"First Counter"=dword:00000842
"First Help"=dword:00000843
"Object List"="2114"
"WbemAdapFileTime"=hex:00,f7,5c,bd,a0,a5,bf,01
"WbemAdapFileSize"=dword:00158910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\isapnp]
"ErrorControl"=dword:00000003
"Group"="Boot Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000003
"Type"=dword:00000001
"HasBootConfig"=dword:00000000
"DisplayName"="PnP ISA/EISA Bus Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,69,73,61,\
  70,6e,70,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\isapnp\Parameters]
"ADP1502"=dword:00000001
"ADP1505"=dword:00000001
"ADP1510"=dword:00000001
"ADP1512"=dword:00000001
"ADP1515"=dword:00000001
"ADP1520"=dword:00000001
"ADP1522"=dword:00000001
"USR0014"=dword:00000001
```

```
"USR1001"=dword:00000001
"USR1002"=dword:00000001
"USR1003"=dword:00000001
"USR1004"=dword:00000001
"USR6001"=dword:00000001
"USR6002"=dword:00000001
"USR6003"=dword:00000001
"USR6004"=dword:00000001
"USR6005"=dword:00000001
"USR6006"=dword:00000001
"USR6007"=dword:00000001
"USR6008"=dword:00000001
"USR6009"=dword:00000001
"USR600A"=dword:00000001
"USR600B"=dword:00000001
"USR600C"=dword:00000001
"USR600D"=dword:00000001
"USR600E"=dword:00000001
"USR600F"=dword:00000001
"USR6010"=dword:00000001
"USR6011"=dword:00000001
"USR6012"=dword:00000001
"USR6101"=dword:00000001
"USR6020"=dword:00000001
"USR0041"=dword:00000001
"USR002C"=dword:00000001
"AZT4029"=dword:00000001
"AZT4023"=dword:00000001
"USR0040"=dword:00000001
"HAY8601"=dword:00000001
"EQX2400"=dword:00000002
"EQX0900"=dword:00000002
"EQX1B00"=dword:00000002
"EQX1700"=dword:00000002
"EQX0700"=dword:00000002
"EQX0F00"=dword:00000002
"EQX0800"=dword:00000002
"EQX1000"=dword:00000002
"EQX3F00"=dword:00000002
"EQX1200"=dword:00000002
"IBM0001"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\isapnp\Enum]
"0"="PCI\\VEN_1166&DEV_0200&SUBSYS_00000000&REV_4F\\3&267a616a&0&78"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\IsmServ]
"DependOnService"=hex(7):53,61,6d,53,53,00,00
"Description"="Allows sending and receiving messages between Windows Advanced Server
sites."
"DisplayName"="Intersite Messaging"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,69,73,6d,73,65,72,76,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000004
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kbdclass]
"ErrorControl"=dword:00000001
"Group"="Keyboard Class"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001
"DisplayName"="Keyboard Class Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6b,62,64,\
  63,6c,61,73,73,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kbdclass\Parameters]
"ConnectMultiplePorts"=dword:00000000
"KeyboardDataQueueSize"=dword:00000064
"KeyboardDeviceBaseName"="KeyboardClass"
"MaximumPortsServiced"=dword:00000003
"SendOutputToAllPorts"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kbdclass\Enum]
"0"="ACPI\\PNP0303\\4&f0b8f99&0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kdc]
"DependOnService"=hex(7):52,70,63,53,73,00,41,66,64,00,00
"Description"="Generates session keys and grants service tickets for mutual
client/server authentication."
"DisplayName"="Kerberos Key Distribution Center"
"ErrorControl"=dword:00000001
"Group"="RemoteValidation"
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000004
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kdc\Security]
"Security"=hex:01,00,14,80,9c,00,00,00,a8,00,00,00,14,00,00,00,34,00,00,00,02,\
  00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,20,02,00,00,02,00,68,00,04,00,00,00,00,03,18,00,8d,00,02,00,01,01,00,\
  00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00,02,00,01,02,00,00,00,\
  00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01,01,00,00,00,00,\
  00,05,04,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KSecDD]
"ErrorControl"=dword:00000001
"Group"="Base"
"Start"=dword:00000000
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\KSecDD\Enum]
"0"="Root\\LEGACY_KSECDD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Server"
"ObjectName"="LocalSystem"
"Description"="Provides RPC support and file, print, and named pipe sharing."
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\AutotunedParamete
rs]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\DefaultSecurity]
"SrvsvcConfigInfo"=hex:01,00,04,80,a0,00,00,00,ac,00,00,00,00,00,00,00,14,00,\
  00,00,02,00,8c,00,05,00,00,00,00,00,1c,00,17,00,0f,00,01,02,00,00,00,00,00,\
  05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,17,00,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,18,00,17,00,0f,00,01,\
  01,00,00,00,00,00,05,12,00,00,00,25,02,00,00,00,00,1c,00,03,00,00,00,01,02,\
  00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00,18,00,01,00,00,\
  00,01,01,00,00,00,00,00,05,0b,00,00,00,23,02,00,00,01,01,00,00,00,00,00,05,\
  12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcConnection"=hex:01,00,04,80,8c,00,00,00,98,00,00,00,00,00,00,00,14,00,\
  00,00,02,00,78,00,04,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,00,00,00,00,\
  05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,01,00,00,00,01,\
  02,00,00,00,00,00,05,20,00,00,00,26,02,00,00,00,00,00,00,00,00,1c,00,01,00,\
  00,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcServerDiskEnum"=hex:01,00,04,80,54,00,00,00,60,00,00,00,00,00,00,00,14,\
  00,00,00,02,00,40,00,02,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,\
  12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcFile"=hex:01,00,04,80,70,00,00,00,7c,00,00,00,00,00,00,00,14,00,00,00,\
  02,00,5c,00,03,00,00,00,00,00,1c,00,11,00,0f,00,01,02,00,00,00,00,00,05,20,\
  00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,11,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,11,00,0f,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,\
  12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcSessionInfo"=hex:01,00,04,80,88,00,00,00,94,00,00,00,00,00,00,00,14,00,\
  00,00,02,00,74,00,04,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,00,00,00,00,\
  05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,01,\
  02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00,18,00,01,00,\
  00,00,01,01,00,00,00,00,00,05,0b,00,00,00,23,02,00,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcShareFileInfo"=hex:01,00,04,80,88,00,00,00,94,00,00,00,00,00,00,00,14,\
  00,00,00,02,00,74,00,04,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,\
  01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00,18,00,01,\
  00,00,00,01,01,00,00,00,00,00,05,0b,00,00,00,23,02,00,00,01,01,00,00,00,00,\
  00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcSharePrintInfo"=hex:01,00,04,80,a4,00,00,00,b0,00,00,00,00,00,00,00,14,\
  00,00,00,02,00,90,00,05,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,13,00,0f,00,\
  01,02,00,00,00,00,00,05,20,00,00,00,26,02,00,00,00,00,00,00,00,00,1c,00,13,\
  00,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00,\
  18,00,01,00,00,00,01,01,00,00,00,00,00,05,0b,00,00,00,23,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcShareAdminInfo"=hex:01,00,04,80,88,00,00,00,94,00,00,00,00,00,00,00,14,\
  00,00,00,02,00,74,00,04,00,00,00,00,00,1c,00,13,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,02,00,00,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,02,00,00,00,\
  01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,00,18,00,01,\
  00,00,00,01,01,00,00,00,00,00,05,0b,00,00,00,23,02,00,00,01,01,00,00,00,00,\
  00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcShareConnect"=hex:01,00,04,80,88,00,00,00,94,00,00,00,00,00,00,00,14,00,\
  00,00,02,00,74,00,04,00,00,00,00,00,1c,00,03,00,0f,00,01,02,00,00,00,00,00,\
  05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,03,00,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,03,00,0f,00,01,\
  02,00,00,00,00,00,05,20,00,00,00,27,02,00,00,00,00,00,00,00,00,18,00,01,00,\
  00,00,01,01,00,00,00,00,01,00,00,00,00,27,02,00,00,01,01,00,00,00,00,00,00,\
  05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcShareAdminConnect"=hex:01,00,04,80,70,00,00,00,7c,00,00,00,00,00,00,00,\
  14,00,00,00,02,00,5c,00,03,00,00,00,00,00,1c,00,03,00,0f,00,01,02,00,00,00,\
  00,00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,03,00,0f,00,01,02,\
  00,00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,1c,00,03,00,0f,\
  00,01,02,00,00,00,00,00,05,20,00,00,00,27,02,00,00,00,00,00,00,01,01,00,00,\
  00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
"SrvsvcStatisticsInfo"=hex:01,00,04,80,6c,00,00,00,78,00,00,00,00,00,00,00,14,\
  00,00,00,02,00,58,00,03,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,00,1c,00,01,00,0f,00,01,02,00,\
  00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,18,00,01,00,00,00,\
  01,01,00,00,00,00,00,02,00,00,00,00,25,02,00,00,01,01,00,00,00,00,00,05,12,\
  00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,62,69,6f,73,53,6d,62,00,5c,44,\
  65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,32,36,32,41,34,45,\
  31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,\
  41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,\
  70,5f,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,\
  36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,63,65,5c,4e,\
  65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,42,2d,37,32,39,31,\
  2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,32,45,34,41,7d,00,\
  00
"Route"=hex(7):22,4e,65,74,62,69,6f,73,53,6d,62,22,00,22,4e,65,74,42,54,22,20,\
  22,54,63,70,69,70,22,20,22,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,\
  36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,22,00,22,\
  4e,65,74,42,54,22,20,22,54,63,70,69,70,22,20,22,4e,64,69,73,57,61,6e,49,70,\
  22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4c,61,6e,6d,61,6e,53,65,72,76,65,72,5f,\
  4e,65,74,62,69,6f,73,53,6d,62,00,5c,44,65,76,69,63,65,5c,4c,61,6e,6d,61,6e,\
  53,65,72,76,65,72,5f,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,32,36,32,41,34,\
  45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,\
  38,41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,4c,61,6e,6d,61,6e,53,65,72,\
  76,65,72,5f,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,32,31,43,44,35,35,37,34,\
  2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,36,2d,45,42,30,46,38,35,36,43,43,\
  38,34,44,7d,00,5c,44,65,76,69,63,65,5c,4c,61,6e,6d,61,6e,53,65,72,76,65,72,\
  5f,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,42,2d,37,32,\
  39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,32,45,34,41,\
  7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters]
"autodisconnect"=dword:0000000f
"enableforcedlogoff"=dword:00000001
"enablesecuritysignature"=dword:00000000
"requiresecuritysignature"=dword:00000000
"NullSessionPipes"=hex(7):43,4f,4d,4e,41,50,00,43,4f,4d,4e,4f,44,45,00,53,51,\
  4c,5c,51,55,45,52,59,00,53,50,4f,4f,4c,53,53,00,4c,4c,53,52,50,43,00,45,50,\
  4d,41,50,50,45,52,00,4c,4f,43,41,54,4f,52,00,54,72,6b,57,6b,73,00,54,72,6b,\
  53,76,72,00,00
"NullSessionShares"=hex(7):43,4f,4d,43,46,47,00,44,46,53,24,00,00
"Lmannounce"=dword:00000000
"Size"=dword:00000003
"Guid"=hex:c5,25,1e,9d,56,c5,3d,45,aa,c4,8d,54,45,41,1a,96

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
```

```
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Shares]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Shares\Security]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver\Enum]
"0"="Root\\LEGACY_LANMANSERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Workstation"
"Group"="NetworkProvider"
"ObjectName"="LocalSystem"
"Description"="Provides network connections and communications."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,62,69,6f,73,53,6d,62,00,5c,44,\
  65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,32,36,32,41,34,45,\
  31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,\
  41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,\
  70,5f,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,\
  36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,63,65,5c,4e,\
  65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,42,2d,37,32,39,31,\
  2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,32,45,34,41,7d,00,\
  00
"Route"=hex(7):22,4e,65,74,62,69,6f,73,53,6d,62,22,00,22,4e,65,74,42,54,22,20,\
  22,54,63,70,69,70,22,20,22,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,\
  36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,22,00,22,\
  4e,65,74,42,54,22,20,22,54,63,70,69,70,22,20,22,4e,64,69,73,57,61,6e,49,70,\
  22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,\
  74,69,6f,6e,5f,4e,65,74,62,69,6f,73,53,6d,62,00,5c,44,65,76,69,63,65,5c,4c,\
  61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,5f,4e,65,74,42,54,5f,54,63,\
  70,69,70,5f,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,\
  45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,\
  5c,4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,5f,4e,65,74,42,54,5f,\
  54,63,70,69,70,5f,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,\
  2d,41,31,35,36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,\
  63,65,5c,4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,5f,4e,65,74,42,\
  54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,42,2d,37,32,39,31,2d,34,34,\
  33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,32,45,34,41,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\NetworkProvi
der]
"Name"="Microsoft Windows Network"
"ProviderPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,6e,74,6c,61,6e,6d,61,6e,2e,64,6c,6c,00
"DeviceName"="\\Device\\LanmanRedirector"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\parameters]
"enableplaintextpassword"=dword:00000000
"enablesecuritysignature"=dword:00000001
"requiresecuritysignature"=dword:00000000
"OtherDomains"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation\Enum]
"0"="Root\\LEGACY_LANMANWORKSTATION\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lbrtfdc]
"ErrorControl"=dword:00000000
"Group"="System Bus Extender"
"Start"=dword:00000001
"Tag"=dword:0000000e
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseInfo]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseInfo\FilePrint]
"Mode"=dword:00000001
"ConcurrentLimit"=dword:0000270f
"FlipAllow"=dword:00000001
"FamilyDisplayName"="Windows Server"
"DisplayName"="Windows Server"
"LocalKey"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,6c,73,73,72,76,2e,65,78,65,00
"DisplayName"="License Logging Service"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint\KSecD
D]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint\MSAfp
Srv]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint\SMBSe
rver]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint\TCP/I
P Print Server]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\FilePrint\TermS
ervice]
@=""
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\Parameters]
"UseEnterprise"=dword:00000000
"ReplicationType"=dword:00000000
"ReplicationTime"=dword:00015180
"EnterpriseServer"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LicenseService\Enum]
"0"="Root\\LEGACY_LICENSESERVICE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LmHosts]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="TCP/IP NetBIOS Helper Service"
"Group"="TDI"
"DependOnService"=hex(7):4e,65,74,42,54,00,41,66,64,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Enables support for NetBIOS over TCP/IP (NetBT) service and NetBIOS
name resolution."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LmHosts\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LmHosts\Enum]
"0"="Root\\LEGACY_LMHOSTS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lp6nds35]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000027
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lp6nds35\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lp6nds35\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Messenger]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="Messenger"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  4e,65,74,42,49,4f,53,00,52,70,63,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Sends and receives messages transmitted by administrators or by the
Alerter service."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Messenger\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Messenger\Enum]
"0"="Root\\LEGACY_MESSENGER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mnmdd]
"ErrorControl"=dword:00000000
"Group"="Video Save"
"Start"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mnmdd\Device0]
"InstalledDisplayDrivers"=hex(7):6d,6e,6d,64,64,00,00
"Device Description"="NetMeeting driver"
"VgaCompatible"=dword:00000000
"MirrorDriver"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mnmdd\Enum]
"0"="Root\\LEGACY_MNMDD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mnmsrvc]
"Type"=dword:00000110
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,6d,6e,\
  6d,73,72,76,63,2e,65,78,65,00
"DisplayName"="NetMeeting Remote Desktop Sharing"
"ObjectName"="LocalSystem"
"Description"="Allows authorized people to remotely access your Windows desktop
using NetMeeting."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mnmsrvc\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,69,00,74,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,79,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
```

```
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,79,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Modem]
"ErrorControl"=dword:00000000
"Group"="Extended base"
"Start"=dword:00000003
"Tag"=dword:00000004
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Modem\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mouclass]
"ErrorControl"=dword:00000001
"Group"="Pointer Class"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001
"DisplayName"="Mouse Class Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6d,6f,75,\
  63,6c,61,73,73,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mouclass\Parameters]
"ConnectMultiplePorts"=dword:00000000
"MaximumPortsServiced"=dword:00000003
"MouseDataQueueSize"=dword:00000064
"PointerDeviceBaseName"="PointerClass"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mouclass\Enum]
"0"="ACPI\\PNP0F13\\4&f0b8f99&0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MountMgr]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000008
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MountMgr\Enum]
"0"="Root\\LEGACY_MOUNTMGR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mraid35x]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000002b
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mraid35x\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mraid35x\Parameters\PnpInterfa
ce]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxSmb]
"Type"=dword:00000002
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000005
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6d,72,78,\
  73,6d,62,2e,73,79,73,00
"DisplayName"="MRXSMB"
"Group"="Network"
"Description"="MRXSMB"
"LastLoadStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxSmb\Parameters]
"CscEnabled"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxSmb\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MRxSmb\Enum]
"0"="Root\\LEGACY_MRXSMB\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSDTC]
"Type"=dword:00000110
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,6d,73,\
  64,74,63,2e,65,78,65,00
"DisplayName"="Distributed Transaction Coordinator"
"Group"="MS Transactions"
"DependOnService"=hex(7):52,50,43,53,53,00,53,61,6d,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Coordinates transactions that are distributed across two or more
databases, message queues, file systems, or other transaction protected resource
managers."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSDTC\Performance]
"Library"="msdtcui.DLL"
"Open"="DtcPerfOpen"
"Collect"="DtcPerfCollect"
"Close"="DtcPerfClose"
"Last Counter"=dword:00000800
"Last Help"=dword:00000801
"First Counter"=dword:000007e6
"First Help"=dword:000007e7
"Object List"="2022"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00023510
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSDTC\Security]
"Security"=hex:01,00,14,80,bc,00,00,00,c8,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,8c,00,06,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,00,00,14,\
```

```
  00,10,00,00,00,01,01,00,00,00,00,00,01,00,00,00,00,01,01,00,00,00,00,00,05,\
  12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSDTC\Enum]
"0"="Root\\LEGACY_MSDTC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Msfs]
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000001
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Msfs\Enum]
"0"="Root\\LEGACY_MSFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSIServer]
"Description"="Installs, repairs and removes software according to instructions
contained in .MSI files."
"Type"=dword:00000120
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,73,79,73,74,65,6d,33,32,5c,6d,73,\
  69,65,78,65,63,2e,65,78,65,20,2f,56,00
"DisplayName"="Windows Installer"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSIServer\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,05,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,12,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSIServer\Enum]
"0"="Root\\LEGACY_MSISERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSKSSRV]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000003
"ImagePath"=hex(2):73,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,4d,53,4b,\
  53,53,52,56,2e,73,79,73,00
"DisplayName"="Microsoft Streaming Service Proxy"
"Group"="ExtendedBase"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSKSSRV\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6c,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSPCLOCK]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"ImagePath"=hex(2):73,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,4d,53,50,\
  43,4c,4f,43,4b,2e,73,79,73,00
"DisplayName"="Microsoft Streaming Clock Proxy"
"Group"="ExtendedBase"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSPCLOCK\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,74,00,6c,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSPQM]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000002
"ImagePath"=hex(2):73,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,4d,53,50,\
  51,4d,2e,73,79,73,00
"DisplayName"="Microsoft Streaming Quality Manager Proxy"
"Group"="ExtendedBase"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSPQM\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLSERVER]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,50,52,4f,47,52,41,7e,31,5c,4d,49,43,52,4f,53,7e,32,\
  5c,4d,53,53,51,4c,5c,62,69,6e,6e,5c,73,71,6c,73,65,72,76,72,2e,65,78,65,00
"DisplayName"="MSSQLSERVER"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLSERVER\Linkage]
"Export"=hex(7):4d,53,53,51,4c,53,45,52,56,45,52,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLSERVER\Performance]
"Library"="C:\\Program Files\\Microsoft SQL Server\\MSSQL\\BINN\\SQLCTR80.DLL"
"Collect"="CollectSQLPerformanceData"
"Open"="OpenSQLPerformanceData"
"Close"="CloseSQLPerformanceData"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLSERVER\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
```

```
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,44,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,44,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLSERVER\Enum]
"0"="Root\\LEGACY_MSSQLSERVER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLServerADHelper]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,50,72,6f,67,72,61,6d,20,46,69,6c,65,73,5c,4d,69,63,\
  72,6f,73,6f,66,74,20,53,51,4c,20,53,65,72,76,65,72,5c,38,30,5c,54,6f,6f,6c,\
  73,5c,42,69,6e,6e,5c,73,71,6c,61,64,68,6c,70,2e,65,78,65,00
"DisplayName"="MSSQLServerADHelper"
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MSSQLServerADHelper\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,47,00,45,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,4e,00,54,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,4e,00,54,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mup]
"DisplayName"="Mup"
"ErrorControl"=dword:00000001
"Group"="Network"
"Start"=dword:00000000
"Tag"=dword:00000002
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mup\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Mup\Enum]
"0"="Root\\LEGACY_MUP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\N100]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:0000000a
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,31,30,\
  30,6e,74,35,2e,73,79,73,00
"DisplayName"="Compaq Ethernet or Fast Ethernet NIC NT Driver"
"Group"="NDIS"
"TextModeFlags"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\N100\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,01,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,c4,02,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,c4,02,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\N100\Enum]
"0"="PCI\\VEN_8086&DEV_1229&SUBSYS_B1440E11&REV_08\\3&13c0b0c5&0&30"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ncrc710]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000005
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS]
"DisplayName"="NDIS System Driver"
"ErrorControl"=dword:00000001
"Group"="NDIS Wrapper"
"Start"=dword:00000000
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\MediaTypes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA]
"BusType"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C592]
"Id"=dword:20596d50
"Mask"=dword:f0ffffff
"Token"="EISA&*TCM5920"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\3C597]
"Id"=dword:70596d50
"Mask"=dword:f0ffffff
"Token"="EISA&*TCM5970"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\579TP]
"Id"=dword:93506d50
"Mask"=dword:ffffffff
"Token"="*TCM5093"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAI]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_TOPNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\BONSAIPORT
2]
"Id"=dword:0062110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6200_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC300]
"Id"=dword:0230a310
"Mask"=dword:0fffffff
"Token"="EISA&DEFEA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DEC422]
```

```
"Id"=dword:2042a310
"Mask"=dword:f0ffffff
"Token"="EISA&DEC422"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\DURANGO]
"Id"=dword:0260110e
"Mask"=dword:0fffffff
"Token"="*CPQ6002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\EE100E]
"Id"=dword:6010d425
"Mask"=dword:f0ffffff
"Token"="EISA&INT1060"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\HP2577A]
"Id"=dword:4019f022
"Mask"=dword:f0ffffff
"Token"="*HWP1940"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\INTEL32E]
"Id"=dword:0112d425
"Mask"=dword:0fffffff
"Token"="*INT1201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MAPLE]
"Id"=dword:0160110e
"Mask"=dword:0fffffff
"Token"="*CPQ6001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MDGEB]
"Id"=dword:03008734
"Mask"=dword:ffffffff
"Token"="*MDG0003"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MDGER1]
"Id"=dword:01008734
"Mask"=dword:ffffffff
"Token"="*MDG0001"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MDGER2]
"Id"=dword:02008734
"Mask"=dword:ffffffff
"Token"="*MDG0002"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\MDGER3]
"Id"=dword:04008734
"Mask"=dword:ffffffff
"Token"="*MDG0004"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NE3200]
"Id"=dword:0007cc3a
"Mask"=dword:00ffffff
"Token"="*NVL0701"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLEX3]
"Id"=dword:20f1110e
"Mask"=dword:f0ffffff
"Token"="*CPQF120"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLEX3.1
]
"Id"=dword:40f1110e
"Mask"=dword:f0ffffff
"Token"="*CPQF140"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NETFLX]
"Id"=dword:0061110e
"Mask"=dword:00ffffff
"Token"="*CPQ6100"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\NPEISA]
"Id"=dword:0000093a
"Mask"=dword:0000ffff
"Token"="*NPIEISA"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\OC3133]
"Id"=dword:0209833d
"Mask"=dword:0fffffff
"Token"="EISA&*OLC0902"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\OCTK32]
"Id"=dword:0112833d
"Mask"=dword:0fffffff
"Token"="*OLC1201"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\P1990]
"Id"=dword:00604f42
"Mask"=dword:00ffffff
"Token"="EISA&*PRO6000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\RODAN]
"Id"=dword:0063110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6300_TOPNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\EISA\RODANPORT2
]
"Id"=dword:0063110e
"Mask"=dword:00ffffff
"Token"="EISA_CPQ6300_BOTTOMNET"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\NetDetect\ISA]
"BusType"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Parameters]
"ProcessorAffinityMask"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDIS\Enum]
"0"="Root\\LEGACY_NDIS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisTapi]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,64,69,\
  73,74,61,70,69,2e,73,79,73,00
"DisplayName"="Remote Access NDIS TAPI Driver"
"Description"="Remote Access NDIS TAPI Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisTapi\Parameters]
"AsyncEventQueueSize"=dword:00000300

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisTapi\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
```

```
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,65,00,72,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisTapi\Enum]
"0"="Root\\MS_NDISWANIP\\0000"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="Root\\MS_PPTPMINIPORT\\0000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisWan]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,64,69,\
  73,77,61,6e,2e,73,79,73,00
"DisplayName"="Remote Access NDIS WAN Driver"
"Description"="Remote Access NDIS WAN Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisWan\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,7b,38,37,36,42,46,38,37,37,2d,33,42,34,\
  41,2d,34,42,38,32,2d,42,35,44,30,2d,36,32,33,46,43,44,41,44,41,36,45,31,7d,\
  00,5c,44,65,76,69,63,65,5c,7b,39,31,45,32,33,32,35,41,2d,36,46,35,36,2d,34,\
  44,41,43,2d,41,43,38,46,2d,36,39,42,35,42,43,34,41,30,46,32,46,7d,00,5c,44,\
  65,76,69,63,65,5c,7b,31,46,36,42,30,44,42,37,2d,30,42,38,38,2d,34,30,44,37,\
  2d,38,33,36,45,2d,34,39,39,37,43,32,42,35,32,30,43,45,7d,00,5c,44,65,76,69,\
  63,65,5c,7b,42,34,36,41,33,36,33,41,2d,46,43,33,45,2d,34,42,30,37,2d,39,36,\
  31,43,2d,37,46,32,38,37,36,39,45,36,36,35,45,7d,00,00
"Route"=hex(7):22,7b,38,37,36,42,46,38,37,37,2d,33,42,34,41,2d,34,42,38,32,2d,\
  42,35,44,30,2d,36,32,33,46,43,44,41,44,41,36,45,31,7d,22,00,22,7b,39,31,45,\
  32,33,32,35,41,2d,36,46,35,36,2d,34,44,41,43,2d,41,43,38,46,2d,36,39,42,35,\
  42,43,34,41,30,46,32,46,7d,22,00,22,7b,31,46,36,42,30,44,42,37,2d,30,42,38,\
  38,2d,34,30,44,37,2d,38,33,36,45,2d,34,39,39,37,43,32,42,35,32,30,43,45,7d,\
  22,00,22,7b,42,34,36,41,33,36,33,41,2d,46,43,33,45,2d,34,42,30,37,2d,39,36,\
  31,43,2d,37,46,32,38,37,36,39,45,36,36,35,45,7d,22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,64,69,73,57,61,6e,5f,7b,38,37,36,42,\
  46,38,37,37,2d,33,42,34,41,2d,34,42,38,32,2d,42,35,44,30,2d,36,32,33,46,43,\
  44,41,44,41,36,45,31,7d,00,5c,44,65,76,69,63,65,5c,4e,64,69,73,57,61,6e,5f,\
  7b,39,31,45,32,33,32,35,41,2d,36,46,35,36,2d,34,44,41,43,2d,41,43,38,46,2d,\
  36,39,42,35,42,43,34,41,30,46,32,46,7d,00,5c,44,65,76,69,63,65,5c,4e,64,69,\
  73,57,61,6e,5f,7b,31,46,36,42,30,44,42,37,2d,30,42,38,38,2d,34,30,44,37,2d,\
  38,33,36,45,2d,34,39,39,37,43,32,42,35,32,30,43,45,7d,00,5c,44,65,76,69,63,\
  65,5c,4e,64,69,73,57,61,6e,5f,7b,42,34,36,41,33,36,33,41,2d,46,43,33,45,2d,\
  34,42,30,37,2d,39,36,31,43,2d,37,46,32,38,37,36,39,45,36,36,35,45,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisWan\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisWan\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NdisWan\Enum]
"0"="Root\\MS_NDISWANIP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDProxy]
"DisplayName"=hex(7):4e,44,49,53,20,50,72,6f,78,79,00,00
"ErrorControl"=dword:00000001
"Group"="PNP_TDI"
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NDProxy\Enum]
"0"="Root\\LEGACY_NDPROXY\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS]
"Type"=dword:00000002
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,65,74,\
  62,69,6f,73,2e,73,79,73,00
"DisplayName"="NetBIOS Interface"
"Group"="NetBIOSGroup"
"Description"="NetBIOS Interface"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS\Linkage]
"LanaMap"=hex:01,00,00,01,00,02
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,\
  32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,\
  31,42,30,45,45,38,41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,4e,65,74,42,\
  54,5f,54,63,70,69,70,5f,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,\
  36,38,2d,41,31,35,36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,\
  76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,\
  42,2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,\
  32,45,34,41,7d,00,00
"Route"=hex(7):22,4e,65,74,42,54,22,20,22,54,63,70,69,70,22,20,22,7b,32,36,32,\
  41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,\
  45,45,38,41,41,35,31,44,7d,22,00,22,4e,65,74,42,54,22,20,22,54,63,70,69,70,\
  22,20,22,4e,64,69,73,57,61,6e,49,70,22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,49,4f,53,5f,4e,65,74,42,54,\
  5f,54,63,70,69,70,5f,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,\
  31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,00,5c,44,65,76,\
  69,63,65,5c,4e,65,74,42,49,4f,53,5f,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,\
  32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,36,2d,45,\
  42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,63,65,5c,4e,65,74,42,\
  49,4f,53,5f,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,42,\
  2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,32,\
  45,34,41,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS\Parameters]
"MaxLana"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS\Parameters\Winsock]
"HelperDllName"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,\
  6d,33,32,5c,77,73,68,6e,65,74,62,73,2e,64,6c,6c,00
"MaxSockAddrLength"=dword:00000014
"MinSockAddrLength"=dword:00000014
"Mapping"=hex:02,00,00,00,03,00,00,00,11,00,00,00,05,00,00,00,00,00,00,00,11,\
  00,00,00,02,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
```

```
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBIOS\Enum]
"0"="Root\\LEGACY_NETBIOS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000005
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,65,74,\
  62,74,2e,73,79,73,00
"DisplayName"="NetBios over Tcpip"
"Group"="PNP_TDI"
"DependOnService"=hex(7):54,63,70,69,70,00,00
"DependOnGroup"=hex(7):00
"Description"="NetBios over Tcpip"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Linkage]
"OtherDependencies"=hex(7):54,63,70,69,70,00,00
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,32,36,32,41,34,45,\
  31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,\
  41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,32,31,43,\
  44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,36,2d,45,42,30,46,\
  38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,\
  30,44,39,32,42,34,31,42,2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,\
  34,46,44,36,39,46,42,32,45,34,41,7d,00,00
"Route"=hex(7):22,54,63,70,69,70,22,20,22,7b,32,36,32,41,34,45,31,32,2d,44,33,\
  39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,\
  7d,22,00,22,54,63,70,69,70,22,20,22,4e,64,69,73,57,61,6e,49,70,22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,\
  32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,\
  31,42,30,45,45,38,41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,4e,65,74,42,\
  54,5f,54,63,70,69,70,5f,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,\
  36,38,2d,41,31,35,36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,5c,44,65,\
  76,69,63,65,5c,4e,65,74,42,54,5f,54,63,70,69,70,5f,7b,30,44,39,32,42,34,31,\
  42,2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,34,46,44,36,39,46,42,\
  32,45,34,41,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters]
"NbProvider"="_tcp"
"NameServerPort"=dword:00000089
"CacheTimeout"=dword:000927c0
"BcastNameQueryCount"=dword:00000003
"BcastQueryTimeout"=dword:000002ee
"NameSrvQueryCount"=dword:00000003
"NameSrvQueryTimeout"=dword:000005dc
"Size/Small/Medium/Large"=dword:00000001
"SessionKeepAlive"=dword:0036ee80
"TransportBindName"="\\Device\\"
"EnableLMHOSTS"=dword:00000001
"DhcpNodeType"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces\Tc
pip_{0D92B41B-7291-443B-B6F1-94FD69FB2E4A}]
"NameServerList"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces\Tc
pip_{21CD5574-1BA5-4968-A156-EB0F856CC84D}]
"NameServerList"=hex(7):00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\Interfaces\Tc
pip_{262A4E12-D39B-46B1-AEA7-F1B0EE8AA51D}]
"NameServerList"=hex(7):00
"NetbiosOptions"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Enum]
"0"="Root\\LEGACY_NETBT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDDE]
"DependOnService"=hex(7):4e,65,74,44,44,45,44,53,44,4d,00,00
"Description"="Provides network transport and security for dynamic data exchange
(DDE)."
"DisplayName"="Network DDE"
"ErrorControl"=dword:00000001
"Group"="NetDDEGroup"
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDDE\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,00,\
  18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDDEdsdm]
"DependOnService"=hex(7):00
"Description"="Manages shared dynamic data exchange and is used by Network DDE"
"DisplayName"="Network DDE DSDM"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,6e,65,74,64,64,65,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDDEdsdm\Security]
```

```
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,00,\
  18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDetect]
"DisplayName"="NetDetect"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5c,73,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,6e,65,74,64,74,65,63,74,2e,73,79,73,00
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetDetect\Enum]
"0"="Root\\LEGACY_NETDETECT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"DisplayName"="Net Logon"
"Group"="RemoteValidation"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Supports pass-through authentication of account logon events for
computers in a domain."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters]
"DisablePasswordChange"=dword:00000000
"requiresignorseal"=dword:00000000
"requirestrongkey"=dword:00000000
"sealsecurechannel"=dword:00000001
"signsecurechannel"=dword:00000001
"DBFlag"="0"
"Update"="no"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netman]
"DependOnService"=hex(7):52,70,63,53,73,00,00
"Description"="Manages objects in the Network and Dial-Up Connections folder, in
which you can view both local area network and remote connections."
"DisplayName"="Network Connections"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000120

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netman\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,6e,65,74,6d,61,6e,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netman\Enum]
"0"="Root\\LEGACY_NETMAN\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Npfs]
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000001
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Npfs\Aliases]
"lsass"=hex(7):6e,65,74,6c,6f,67,6f,6e,00,6c,73,61,72,70,63,00,73,61,6d,72,00,\
  00
"ntsvcs"=hex(7):73,72,76,73,76,63,00,77,6b,73,73,76,63,00,65,76,65,6e,74,6c,6f,\
  67,00,62,72,6f,77,73,65,72,00,6d,73,67,73,76,63,00,73,76,63,63,74,6c,00,77,\
  33,32,74,69,6d,65,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Npfs\Enum]
"0"="Root\\LEGACY_NPFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs]
"DependOnService"=hex(7):45,76,65,6e,74,4c,6f,67,00,52,70,63,53,73,00,00
"Description"="Maintains file synchronization of file directory contents among
multiple servers."
"DisplayName"="File Replication"
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,6e,74,66,72,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters]
"Working Directory"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,6e,74,66,72,\
  73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtFrs\Parameters\SysVol]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ntfs]
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000004
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ntfs\Enum]
"0"="Root\\LEGACY_NTFS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtLmSsp]
"Type"=dword:00000020
```

```
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"DisplayName"="NT LM Security Support Provider"
"ObjectName"="LocalSystem"
"Description"="Provides security to remote procedure call (RPC) programs that use
transports other than named pipes."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtLmSsp\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc]
"DependOnService"=hex(7):52,70,63,53,73,00,00
"Description"="Manages removable media, drives, and libraries."
"DisplayName"="Removable Storage"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000120

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc\Config]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc\Config\Standalone]
"DriveList"="CdRom0"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,4e,74,6d,73,53,76,63,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NtmsSvc\Enum]
"0"="Root\\LEGACY_NTMSSVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Null]
"ErrorControl"=dword:00000001
"Group"="Base"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Null\Enum]
"0"="Root\\LEGACY_NULL\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NwlnkFlt]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,77,6c,\
  6e,6b,66,6c,74,2e,73,79,73,00
"DisplayName"="IPX Traffic Filter Driver"
"DependOnService"=hex(7):4e,77,6c,6e,6b,46,77,64,00,00
"DependOnGroup"=hex(7):00
"Description"="IPX Traffic Filter Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NwlnkFlt\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NwlnkFwd]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,6e,77,6c,\
  6e,6b,66,77,64,2e,73,79,73,00
"DisplayName"="IPX Traffic Forwarder Driver"
"Description"="IPX Traffic Forwarder Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NwlnkFwd\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parallel]
"DependOnGroup"=hex(7):50,61,72,61,6c,6c,65,6c,20,61,72,62,69,74,72,61,74,6f,\
  72,00,00
"DependOnService"=hex(7):50,61,72,70,6f,72,74,00,00
"ErrorControl"=dword:00000001
"Group"="Extended base"
"Start"=dword:00000003
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="Parallel class driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,70,61,72,\
  61,6c,6c,65,6c,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parallel\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parallel\Enum]
"0"="Root\\PARALLELCLASS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parport]
"ErrorControl"=dword:00000000
"Group"="Parallel arbitrator"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001
"DisplayName"="Parallel port driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,70,61,72,\
  70,6f,72,74,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Parport\Enum]
"0"="ACPI\\PNP0400\\5&24f21e77&0"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PartMgr]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000005
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PartMgr\Enum]
"0"="Root\\LEGACY_PARTMGR\\0000"
"Count"=dword:0000000c
"NextInstance"=dword:0000000c
"1"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&000"
"2"="SCSI\\Disk&Ven_Compaq&Prod_Disk_Array&Rev_1.42\\4&20c2a8c9&0&010"
"3"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0000004000000000"
"4"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0100004000000000"
"5"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0200004000000000"
"6"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0000004000000000"
"7"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0100004000000000"
"8"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0200004000000000"
"9"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0000004000000000"
"10"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0100004000000000"
"11"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0200004000000000"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ParVdm]
"DependOnGroup"=hex(7):50,61,72,61,6c,6c,65,6c,20,61,72,62,69,74,72,61,74,6f,\
  72,00,00
"DependOnService"=hex(7):50,61,72,70,6f,72,74,00,00
"ErrorControl"=dword:00000000
"Group"="Extended base"
"Start"=dword:00000002
"Tag"=dword:00000002
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ParVdm\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ParVdm\Enum]
"0"="Root\\LEGACY_PARVDM\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCI]
"ErrorControl"=dword:00000003
"Group"="Boot Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000002
"Type"=dword:00000001
"DisplayName"="PCI Bus Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,70,63,69,\
  2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCI\Parameters]
"Options"=dword:00000002
"C6211045"=dword:00000004
"06401095"=dword:00000004
"12308086"=dword:00000004
"70108086"=dword:00000004
"0140104B"=dword:00000008
"06031179"=dword:00000008
"71138086"=dword:00000008
"84C54978"=dword:00000008
"30401106"=dword:00000008
"10000E11"=dword:00000010
"20000E11"=dword:00000010
"04061039"=dword:00000010
"04828086"=dword:00004000
"00088086"=dword:00000010
"00021014"=dword:00000010
"06001080"=dword:00000020
"11001013"=dword:00000040
"521910B9"=dword:00000080
"00011C1C"=dword:00000100
"00381097"=dword:00000100
"D001100B"=dword:00000400
"04A38086"=dword:00000800
"000010AA"=dword:00000800
"88D15333"=dword:00000800
"06051179"=dword:00001000
"11101013"=dword:00002000
"04781180"=dword:00002000
"04751180"=dword:00002000
"04761180"=dword:00002000
"01011004"=dword:00004000
"10001042"=dword:00004000
"AC12104C"=dword:00010000
"04661180"=dword:00010000
"00951014"=dword:00040000
"24188086"=dword:00040000
"24288086"=dword:00040000
"244E8086"=dword:00040000
"24488086"=dword:00040000
"122E8086"=dword:00080000
"70008086"=dword:00080000
"71108086"=dword:00080000
"76008086"=dword:00080000
"47471002"=dword:00400000
"47541002"=dword:00400000
"89015333"=dword:00400000
"00D61013"=dword:00400000
"AC15104C"=dword:00400000
"0004110B"=dword:00400000
"000F1000"=dword:00400000
"AC17104C"=dword:00400000
"93971023"=dword:00400000
"47421002"=dword:00400000
"47441002"=dword:00400000
"47491002"=dword:00400000
"47501002"=dword:00400000
"47511002"=dword:00400000
"47551002"=dword:00400000
"47571002"=dword:20400000
"47591002"=dword:20400000
```

```
"4C421002"=dword:00400000
"4C441002"=dword:00400000
"4C471002"=dword:00400000
"4C491002"=dword:00400000
"4C501002"=dword:00400000
"4C511002"=dword:00400000
"56541002"=dword:00400000
"56551002"=dword:00400000
"56561002"=dword:00400000
"0003121A"=dword:00400000
"12318086"=dword:01000000
"00021273"=dword:01000000
"007D1014"=dword:01000000
"01001285"=dword:01000000
"68361217"=dword:08000000
"68321217"=dword:08000000
"07A01091"=dword:20000000
"78008086"=dword:20000000
"800510c8"=dword:20000000
"800610c8"=dword:20000000
"000510c8"=dword:20000000
"000610c8"=dword:20000000
"1001102B"=dword:80000000
"06461095"=dword:20000000
"06701095"=dword:20000000
"00261011"=dword:20000000
"B1548086"=dword:20000000
"89045333"=dword:20000000
"06091179"=dword:40000000
"00471014"=dword:40000000
"051B102B"=dword:80000000
"0520102B"=dword:80000000
"1025102B"=dword:80000000
"0525102B"=dword:80000000
"71218086"=dword:80000000
"71238086"=dword:80000000
"71258086"=dword:80000000
"11328086"=dword:80000000
"00509005"=dword:80000000
"005F9005"=dword:80000000
"4C4D1002"=dword:80000000
"4C521002"=dword:80000000
"4C4E1002"=dword:80000000
"4C531002"=dword:80000000
"47521002"=dword:80000000
"474F1002"=dword:80000000
"474D1002"=dword:80000000
"47531002"=dword:80000000
"474C1002"=dword:80000000
"474E1002"=dword:80000000
"98801023"=dword:80000000
"00A010DE"=dword:80000000
"00A110DE"=dword:80000000
"00A310DE"=dword:80000000
"00B010DE"=dword:80000000
"00B110DE"=dword:80000000
"00B310DE"=dword:80000000
"010010DE"=dword:80000000
"010110DE"=dword:80000000
"010210DE"=dword:80000000
"010310DE"=dword:80000000
"012010DE"=dword:80000000
"012110DE"=dword:80000000
"012210DE"=dword:80000000
"012310DE"=dword:80000000
"015010DE"=dword:80000000
"015110DE"=dword:80000000
"015210DE"=dword:80000000
"015310DE"=dword:80000000
"020010DE"=dword:80000000
"020110DE"=dword:80000000
"020210DE"=dword:80000000
"020310DE"=dword:80000000
"001812D2"=dword:80000000
"001912D2"=dword:80000000
"0521102B"=dword:a0000000
"3202100C"=dword:00008a00
"80021066"=dword:00300000
"00021066"=dword:00300000
"01021004"=dword:02004000
"C8141045"=dword:20400000
"47561002"=dword:20400000
"475A1002"=dword:20400000
"000B1000"=dword:a0000000
"002010DE"=dword:a0000000
"002810DE"=dword:a0000000
"002910DE"=dword:a0000000
"002A10DE"=dword:a0000000
"002B10DE"=dword:a0000000
"002C10DE"=dword:a0000000
"002D10DE"=dword:a0000000
"002E10DE"=dword:a0000000
"002F10DE"=dword:a0000000
"010010DD"=dword:20000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCI\Enum]
"0"="ACPI\\PNP0A03\\0"
"Count"=dword:00000003
"NextInstance"=dword:00000003
"1"="ACPI\\PNP0A03\\1"
"2"="ACPI\\PNP0A03\\2"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCIDump]
"ErrorControl"=dword:00000000
"Group"="PCI Configuration"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCIIde]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
"Start"=dword:00000000
"Tag"=dword:00000003
"Type"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,70,63,69,\
  69,64,65,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PCIIde\Enum]
"0"="PCI\\VEN_1166&DEV_0211&SUBSYS_00000000&REV_00\\3&267a616a&0&79"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Pcmcia]
"ErrorControl"=dword:00000001
"Group"="System Bus Extender"
```

```
"Start"=dword:00000004
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Pcmcia\Parameters]
"ControllerInterruptMask"=dword:0000ffff
"Interrupt"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PDCOMP]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PDFRAME]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PDRELI]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PDRFRAME]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfDisk]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfDisk\Performance]
"Close"="CloseDiskObject"
"Collect"="CollectDiskObjectData"
"Collect Timeout"=dword:000007d0
"Library"="perfdisk.dll"
"Object List"="234 236"
"Open"="OpenDiskObject"
"Open Timeout"=dword:00001388
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00005910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfNet]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfNet\Performance]
"Close"="CloseNetSvcsObject"
"Collect"="CollectNetSvcsObjectData"
"Collect Timeout"=dword:00001388
"Library"="perfnet.dll"
"Object List"="52 262 330 1300"
"Open"="OpenNetSvcsObject"
"Open Timeout"=dword:00001f40
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00004510
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfOS]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfOS\Performance]
"Close"="CloseOSObject"
"Collect"="CollectOSObjectData"
"Collect Timeout"=dword:000007d0
"Library"="perfos.dll"
"Object List"="2 4 86 238 260 700"
"Open"="OpenOSObject"
"Open Timeout"=dword:00001388
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00005510
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfProc]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PerfProc\Performance]
"Close"="CloseSysProcessObject"
"Collect"="CollectSysProcessObjectData"
"Collect Timeout"=dword:00001f40
"Library"="perfproc.dll"
"Object List"="230 232 786 740 816 1408 1500 1548"
"Open"="OpenSysProcessObject"
"Open Timeout"=dword:00002710
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00006d10
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PlugPlay]
"Description"="Manages device installation and configuration and notifies programs
of device changes."
"DisplayName"="Plug and Play"
"ErrorControl"=dword:00000001
"Group"="PlugPlay"
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"PlugPlayServiceType"=dword:00000003
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PlugPlay\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"DisplayName"="IPSEC Policy Agent"
"DependOnService"=hex(7):52,50,43,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Manages IP security policy and starts the ISAKMP/Oakley (IKE) and the
IP security driver."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
```

```
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent\Enum]
"0"="Root\\LEGACY_POLICYAGENT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PptpMiniport]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,61,73,\
  70,70,74,70,2e,73,79,73,00
"DisplayName"="WAN Miniport (PPTP)"
"Description"="WAN Miniport (PPTP)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PptpMiniport\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,fe,2e,d7,29,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,fe,2e,d7,29,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PptpMiniport\Enum]
"0"="Root\\MS_PPTPMINIPORT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ProtectedStorage]
"DependOnService"=hex(7):52,70,63,53,73,00,00
"Description"="Provides protected storage for sensitive data, such as private keys,
to prevent access by unauthorized services, processes, or users."
"DisplayName"="Protected Storage"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000120

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ProtectedStorage\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ProtectedStorage\Enum]
"0"="Root\\LEGACY_PROTECTEDSTORAGE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ptilink]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,70,74,69,\
  6c,69,6e,6b,2e,73,79,73,00
"DisplayName"="Direct Parallel Link Driver"
"Description"="Direct Parallel Link Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ptilink\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,98,3a,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,98,3a,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ptilink\Enum]
"0"="Root\\MS_PTIMINIPORT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1080]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000003d
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1080\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1080\Parameters\PnpInterface
]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ql10wnt]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000023
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ql10wnt\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Ql10wnt\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1240]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000031
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1240\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql1240\Parameters\PnpInterface
]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2100]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
```

```
"Start"=dword:00000004
"Tag"=dword:00000028
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2100\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2100\Parameters\PnpInterface
]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAcd]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,61,73,\
  61,63,64,2e,73,79,73,00
"DisplayName"="Remote Access Auto Connection Driver"
"Group"="Streams Drivers"
"Description"="Remote Access Auto Connection Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAcd\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAcd\Enum]
"0"="Root\\LEGACY_RASACD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAuto]
"Type"=dword:00000120
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"DisplayName"="Remote Access Auto Connection Manager"
"DependOnService"=hex(7):52,61,73,4d,61,6e,00,54,61,70,69,73,72,76,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Creates a connection to a remote network whenever a program
references a remote DNS or NetBIOS name or address."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAuto\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,72,61,73,61,75,74,6f,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasAuto\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rasl2tp]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,61,73,\
  6c,32,74,70,2e,73,79,73,00
"DisplayName"="WAN Miniport (L2TP)"
"Description"="WAN Miniport (L2TP)"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rasl2tp\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,bd,fb,d9,ef,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,fe,2e,d7,29,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,fe,2e,d7,29,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rasl2tp\Enum]
"0"="Root\\MS_L2TPMINIPORT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan]
"Type"=dword:00000120
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"DisplayName"="Remote Access Connection Manager"
"DependOnService"=hex(7):54,61,70,69,73,72,76,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Creates a network connection."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Parameters]
"Medias"=hex(7):72,61,73,74,61,70,69,00,00
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,72,61,73,6d,61,6e,73,2e,64,6c,6c,00
"IpOutLowWatermark"=dword:00000001
"IpOutHighWatermark"=dword:00000005

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP]
"MaxConfigure"=dword:0000000a
"MaxFailure"=dword:0000000a
"MaxReject"=dword:00000005
"MaxTerminate"=dword:00000002
"Multilink"=dword:00000000
"NegotiateTime"=dword:00000096
"RestartTimer"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\ControlProtocols]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\ControlProtocols\Bu
iltIn]
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,70,70,70,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\ControlProtocols\Ch
ap]
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,63,68,61,70,2e,64,6c,6c,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP]
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,70,70,70,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\13]
"FriendlyName"="Smart Card or other Certificate"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,74,6c,73,2e,64,6c,6c,00
"ConfigUiPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,72,61,73,74,6c,73,2e,64,6c,6c,00
"IdentityPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,72,61,73,74,6c,73,2e,64,6c,6c,00
"InteractiveUIPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,72,61,73,74,6c,73,2e,64,6c,6c,00
"InvokeUsernameDialog"=dword:00000000
"InvokePasswordDialog"=dword:00000000
"MPPEEncryptionSupported"=dword:00000001
"ConfigCLSID"="{58AB2366-D597-11d1-B90E-00C04FC9B263}"
"StandaloneSupported"=dword:00000000
"NoRootRevocationCheck"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\PPP\EAP\4]
"FriendlyName"="MD5-Challenge"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,63,68,61,70,2e,64,6c,6c,00
"InvokeUsernameDialog"=dword:00000001
"InvokePasswordDialog"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Security]
"Security"=hex:01,00,14,80,7c,00,00,00,88,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,4c,00,03,00,00,00,00,00,14,00,9d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,\
  00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Enum]
"0"="Root\\LEGACY_RASMAN\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Raspti]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,61,73,\
  70,74,69,2e,73,79,73,00
"DisplayName"="Direct Parallel"
"Description"="Direct Parallel"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Raspti\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,98,3a,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,98,3a,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Raspti\Enum]
"0"="Root\\MS_PTIMINIPORT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RCA]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000004
"ImagePath"=hex(2):73,79,73,74,65,6d,33,32,5c,64,72,69,76,65,72,73,5c,52,43,41,\
  2e,73,79,73,00
"DisplayName"="Microsoft Streaming Network Raw Channel Access"
"Group"="ExtendedBase"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RCA\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdbss]
"Type"=dword:00000002
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000004
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,64,62,\
  73,73,2e,73,79,73,00
"DisplayName"="Rdbss"
"Group"="Network"
"Description"="Rdbss"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdbss\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdbss\Enum]
"0"="Root\\LEGACY_RDBSS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RDPDD]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RDPDD\Device0]
"InstalledDisplayDrivers"=hex(7):52,44,50,44,44,00,00
"VgaCompatible"=dword:00000000
"DefaultSettings.XResolution"=dword:00000320
"DefaultSettings.YResolution"=dword:00000258
"Attach.RelativeX"=dword:00000000
"Attach.RelativeY"=dword:00000000
"Attach.ToDesktop"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\rdpdr]
"Type"=dword:00000001
```

```
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,64,70,\
  64,72,2e,73,79,73,00
"DisplayName"="Terminal Server Device Redirector Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\rdpdr\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\rdpdr\Enum]
"0"="Root\\RDPDR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RDPWD]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RDPWD\Enum]
"0"="Root\\LEGACY_RDPWD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\redbook]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000002
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,72,65,64,\
  62,6f,6f,6b,2e,73,79,73,00
"DisplayName"="Digital CD Audio Playback Filter Driver"
"Group"="Pnp Filter"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\redbook\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,69,00,76,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,65,00,72,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,65,00,72,00,01,01,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\redbook\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000
"INITSTARTFAILED"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess]
"Type"=dword:00000120
"Start"=dword:00000004
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"DisplayName"="Routing and Remote Access"
"DependOnService"=hex(7):52,70,63,53,53,00,00
"DependOnGroup"=hex(7):4e,65,74,42,49,4f,53,47,72,6f,75,70,00,00
"ObjectName"="LocalSystem"
"Description"="Offers routing services to businesses in local area and wide area
network environments."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Accounting]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Accounting\Provid
ers]
"ActiveProvider"="{1AA7F846-C7F5-11D0-A376-00C04FC9DA04}"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Accounting\Provid
ers\{1AA7F840-C7F5-11D0-A376-00C04FC9DA04}]
"ConfigClsid"="{1AA7F840-C7F5-11D0-A376-00C04FC9DA04}"
"DisplayName"="RADIUS Accounting"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,72,61,64,2e,64,6c,6c,00
"ProviderTypeGUID"="{76560D80-2BFD-11d2-9539-3078302C2030}"
"VendorName"="Microsoft"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Accounting\Provid
ers\{1AA7F846-C7F5-11D0-A376-00C04FC9DA04}]
"ConfigClsid"=""
"DisplayName"="Windows Accounting"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,61,75,74,68,2e,64,6c,6c,00
"ProviderTypeGUID"="{76560D81-2BFD-11d2-9539-3078302C2030}"
"VendorName"="Microsoft"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Authentication]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Authentication\Pr
oviders]
"ActiveProvider"="{1AA7F841-C7F5-11D0-A376-00C04FC9DA04}"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Authentication\Pr
oviders\{1AA7F83F-C7F5-11D0-A376-00C04FC9DA04}]
"ConfigClsid"="{1AA7F83F-C7F5-11D0-A376-00C04FC9DA04}"
"DisplayName"="RADIUS Authentication"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,72,61,64,2e,64,6c,6c,00
"VendorName"="Microsoft"
"ProviderTypeGUID"="{76560D00-2BFD-11d2-9539-3078302C2030}"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Authentication\Pr
oviders\{1AA7F841-C7F5-11D0-A376-00C04FC9DA04}]
"ConfigClsid"=""
"DisplayName"="Windows Authentication"
"Path"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,32,\
  5c,72,61,73,61,75,74,68,2e,64,6c,6c,00
"VendorName"="Microsoft"
"ProviderTypeGUID"="{76560D01-2BFD-11d2-9539-3078302C2030}"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\DemandDialManager
]
"DllPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6d,70,72,64,64,6d,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces]
"Stamp"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\0]
```

```
"InterfaceName"="Loopback"
"Type"=dword:00000005
"Enabled"=dword:00000001
"Stamp"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\0\Ip]
"ProtocolId"=dword:00000021
"InterfaceInfo"=hex:01,00,00,00,68,00,00,00,03,00,00,00,05,00,ff,ff,38,00,00,\
  00,00,00,00,00,40,00,00,00,04,00,ff,ff,04,00,00,00,01,00,00,00,40,00,00,00,\
  07,00,ff,ff,10,00,00,00,01,00,00,00,48,00,00,00,00,00,00,00,01,00,00,00,00,\
  00,00,00,58,02,c2,01,08,07,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\1]
"InterfaceName"="Internal"
"Type"=dword:00000004
"Enabled"=dword:00000001
"Stamp"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\1\Ip]
"ProtocolId"=dword:00000021
"InterfaceInfo"=hex:01,00,00,00,68,00,00,00,03,00,00,00,05,00,ff,ff,38,00,00,\
  00,00,00,00,00,40,00,00,00,04,00,ff,ff,04,00,00,00,01,00,00,00,40,00,00,00,\
  07,00,ff,ff,10,00,00,00,01,00,00,00,48,00,00,00,00,00,00,00,01,00,00,00,00,\
  00,00,00,58,02,c2,01,08,07,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\2]
"InterfaceName"="{262A4E12-D39B-46B1-AEA7-F1B0EE8AA51D}"
"Type"=dword:00000003
"Enabled"=dword:00000001
"Stamp"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Interfaces\2\Ip]
"ProtocolId"=dword:00000021
"InterfaceInfo"=hex:01,00,00,00,68,00,00,00,03,00,00,00,05,00,ff,ff,38,00,00,\
  00,00,00,00,00,40,00,00,00,04,00,ff,ff,04,00,00,00,01,00,00,00,40,00,00,00,\
  07,00,ff,ff,10,00,00,00,01,00,00,00,48,00,00,00,00,00,00,00,01,00,00,00,00,\
  00,00,00,58,02,c2,01,08,07,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters]
"RouterType"=dword:00000001
"ServerFlags"=dword:00402702
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,6d,70,72,64,69,6d,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\AppleT
alk]
"EnableIn"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\Ip]
"AllowClientIpAddresses"=dword:00000000
"AllowNetworkAccess"=dword:00000001
"EnableIn"=dword:00000001
"IpAddress"="0.0.0.0"
"IpMask"="0.0.0.0"
"UseDhcpAddressing"=dword:00000001
"EnableRoute"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\Ipx]
"EnableIn"=dword:00000001
"AcceptRemoteNodeNumber"=dword:00000001
"AllowNetworkAccess"=dword:00000001
"AutoWanNetAllocation"=dword:00000001
"FirstWanNet"=dword:00000000
"GlobalWanNet"=dword:00000001
"LastWanNet"=dword:00000000
"EnableRoute"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Parameters\Nbf]
"EnableIn"=dword:00000001
"AllowNetworkAccess"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Performance]
"Open"="OpenRasPerformanceData"
"Close"="CloseRasPerformanceData"
"Collect"="CollectRasPerformanceData"
"Library"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,72,61,73,63,74,72,73,2e,64,6c,6c,00
"Last Counter"=dword:000007e4
"Last Help"=dword:000007e5
"First Counter"=dword:000007be
"First Help"=dword:000007bf
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00003110
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy]
"ProductDir"="C:\\WINNT\\System32\\IAS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01]
"Description"="Begin Session Pipeline"
"Events"=""
"Id"=dword:00000001
"Merit"=dword:00000001
"Name"="Begin Session"
"ParentId"=dword:00000000
"ProgId"="IAS.Pipeline"
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\01]
"Description"="Roaming (Realms) Stage"
"Events"=""
"Id"=dword:00000001
"Merit"=dword:00000001
"Name"="Roaming"
"ParentId"=dword:00000001
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\01\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\01\RequestHandlers\12]
"Description"="Microsoft Names Provider"
"Events"="1"
"Id"=dword:0000000c
```

```
"Merit"=dword:00000001
"Name"="MS Names"
"ParentId"=dword:00000003
"ProgId"="IAS.NTSamNames"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\02]
"Description"="User Credential Retrieval"
"Events"=""
"Id"=dword:00000002
"Merit"=dword:00000002
"Name"="Credential Retrieval"
"ParentId"=dword:00000001
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\02\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\02\RequestHandlers\01]
"Description"="Microsoft Credential Retrieval - NT SAM"
"Events"="1"
"Id"=dword:00000001
"Merit"=dword:00000002
"Name"="Authentication"
"ParentId"=dword:00000004
"ProgId"="IAS.NTSamAuthentication"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\02\RequestHandlers\03]
"Description"="Microsoft BaseCamp Extension Host"
"Events"="1 4"
"Id"=dword:00000003
"Merit"=dword:00000001
"Name"="BaseCamp Host"
"ParentId"=dword:00000004
"ProgId"="IAS.BaseCampHost"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\03]
"Description"="Network Access Policy Evaluation"
"Events"=""
"Id"=dword:00000003
"Merit"=dword:00000003
"Name"="NAP Evaluation"
"ParentId"=dword:00000001
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\03\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\03\RequestHandlers\07]
"Description"="Microsoft Network Access Policy Evaluator"
"Events"="1 2"
"Id"=dword:00000007
"Merit"=dword:00000001
"Name"="MS NAP"
"ParentId"=dword:00000005
"ProgId"="IAS.PolicyEnforcer"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\04]
"Description"="User Authorization"
"Events"=""
"Id"=dword:00000004
"Merit"=dword:00000004
"Name"="User Authorization"
"ParentId"=dword:00000001
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\04\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
01\Stages\04\RequestHandlers\06]
"Description"="Microsoft NT SAM User Authorizations"
"Events"="1 2"
"Id"=dword:00000006
"Merit"=dword:00000001
"Name"="MS User Authorizations"
"ParentId"=dword:00000006
"ProgId"="IAS.NTSamPerUser"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
02]
"Description"="In Session Pipeline"
"Events"=""
"Id"=dword:00000002
"Merit"=dword:00000002
"Name"="In Session"
"ParentId"=dword:00000000
"ProgId"="IAS.Pipeline"
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
02\Stages]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
02\Stages\05]
"Description"="User Authentication Stage"
"Events"=""
"Id"=dword:00000005
"Merit"=dword:00000001
"Name"="User Authentication"
"ParentId"=dword:00000002
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
02\Stages\05\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
02\Stages\05\RequestHandlers\10]
"Description"="Microsoft EAP Authentication DLL Wrapper"
"Events"="1"
"Id"=dword:0000000a
"Merit"=dword:00000001
```

```
"Name"="MS EAP DLL Wrapper"
"ParentId"=dword:00000007
"ProgId"="IAS.EAP"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03]
"Description"="End Session Pipeline"
"Events"=""
"Id"=dword:00000003
"Merit"=dword:00000003
"Name"="End Session"
"ParentId"=dword:00000000
"ProgId"="IAS.Pipeline"
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\06]
"Description"="User Restrictions Stage"
"Events"=""
"Id"=dword:00000006
"Merit"=dword:00000001
"Name"="User Restrictions"
"ParentId"=dword:00000003
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\06\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\06\RequestHandlers\04]
"Description"="Microsoft Authorization Extension Host"
"Events"="1 4"
"Id"=dword:00000004
"Merit"=dword:00000002
"Name"="Authorization Host"
"ParentId"=dword:00000009
"ProgId"="IAS.AuthorizationHost"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\06\RequestHandlers\11]
"Description"="Microsoft User Restrictions Handler DLL"
"Events"="1"
"Id"=dword:0000000b
"Merit"=dword:00000001
"Name"="IAS User Restrictions Handler"
"ParentId"=dword:00000009
"ProgId"="IAS.URHandler"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\07]
"Description"="Logging Stage"
"Events"=""
"Id"=dword:00000007
"Merit"=dword:00000002
"Name"="Logging"
"ParentId"=dword:00000003
"ProgId"="IAS.Stage"
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\07\RequestHandlers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\07\RequestHandlers\02]
"Description"="Microsoft MS-CHAP Error Reporter"
"Events"="1"
"Id"=dword:00000002
"Merit"=dword:00000001
"Name"="MS-CHAP Error Reporter"
"ParentId"=dword:00000008
"ProgId"="IAS.MSChapErrorReporter"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Policy\Pipelines\
03\Stages\07\RequestHandlers\09]
"Description"="Microsoft Logging Provider"
"Events"="1 4"
"Id"=dword:00000009
"Merit"=dword:00000002
"Name"="MS Logging"
"ParentId"=dword:00000008
"ProgId"="IAS.Accounting"
"Type"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\RouterManagers]
"Stamp"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\RouterManagers\Ip
]
"ProtocolId"=dword:00000021
"GlobalInfo"=hex:01,00,00,00,80,00,00,00,02,00,00,00,03,00,ff,ff,08,00,00,00,\
  01,00,00,00,30,00,00,00,06,00,ff,ff,3c,00,00,00,01,00,00,00,38,00,00,00,00,\
  00,00,00,00,00,00,01,00,00,00,07,00,00,00,02,00,00,00,01,00,00,00,03,00,\
  00,00,0a,00,00,00,16,27,00,00,03,00,00,00,17,27,00,00,05,00,00,00,12,27,00,\
  00,07,00,00,00,0d,00,00,00,6e,00,00,00,08,00,00,00,78,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00
"DLLPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,69,70,72,74,72,6d,67,72,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteAccess\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,bd,fb,d9,ef,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,fe,2e,d7,29,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,fe,2e,d7,29,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteRegistry]
"Description"="Allows remote registry manipulation."
"DisplayName"="Remote Registry Service"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,72,65,67,73,76,63,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000010
"FailureActions"=hex:00,00,00,00,00,00,00,00,00,00,00,00,01,00,00,00,e0,ad,08,\
```

```
  00,01,00,00,00,e8,03,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteRegistry\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RemoteRegistry\Enum]
"0"="Root\\LEGACY_REMOTEREGISTRY\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcLocator]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,6c,6f,63,61,74,6f,72,2e,65,78,65,00
"DisplayName"="Remote Procedure Call (RPC) Locator"
"DependOnService"=hex(7):4c,61,6e,6d,61,6e,57,6f,72,6b,73,74,61,74,69,6f,6e,00,\
  00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Manages the RPC name service database."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcLocator\Parameters]
"ExpirationAge"=dword:00000e10

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcLocator\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs]
"Description"="Provides the endpoint mapper and other miscellaneous RPC services."
"DisplayName"="Remote Procedure Call (RPC)"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,20,2d,6b,20,72,70,63,73,73,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,\
  33,32,5c,72,70,63,73,73,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,03,\
  18,00,9d,00,00,00,01,02,00,00,00,00,00,05,20,00,00,00,21,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RpcSs\Enum]
"0"="Root\\LEGACY_RPCSS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RSVP]
"Type"=dword:00000110
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,72,73,76,70,2e,65,78,65,20,2d,73,00
"DisplayName"="QoS RSVP"
"DependOnService"=hex(7):54,63,70,49,70,00,41,66,64,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides network signaling and local traffic control setup
functionality for QoS-aware programs and control applets."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RSVP\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RSVP\Performance]
"Open"="OpenRsvpPerformanceData"
"Close"="CloseRsvpPerformanceData"
"Collect"="CollectRsvpPerformanceData"
"Library"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,72,73,76,70,70,65,72,66,2e,64,6c,6c,00
"Last Counter"=dword:000007bc
"Last Help"=dword:000007bd
"First Counter"=dword:00000738
"First Help"=dword:00000739
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00002b10
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RSVP\Security]
"Security"=hex:01,00,14,80,7c,00,00,00,88,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,4c,00,03,00,00,00,00,00,14,00,9d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,\
  00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SamSs]
"Description"="Stores security information for local user accounts."
"DisplayName"="Security Accounts Manager"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,6c,73,61,73,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SamSs\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
```

```
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,03,\
  18,00,9d,00,00,00,01,02,00,00,00,00,00,05,20,00,00,00,21,02,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SamSs\Enum]
"0"="Root\\LEGACY_SAMSS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardDrv]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,53,43,61,72,64,53,76,72,2e,65,78,65,00
"Description"="Provides support for legacy smart card readers attached to the
computer."
"DisplayName"="Smart Card Helper"
"DependOnGroup"=hex(7):53,6d,61,72,74,20,43,61,72,64,20,52,65,61,64,65,72,00,\
  00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardDrv\Enum]
"0"="Root\\LEGACY_SCARDDRV\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardSvr]
"Type"=dword:00000020
"Start"=dword:00000003
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,53,43,61,72,64,53,76,72,2e,65,78,65,00
"Description"="Manages and controls access to a smart card inserted into a smart
card reader attached to the computer."
"DisplayName"="Smart Card"
"DependOnService"=hex(7):50,6c,75,67,50,6c,61,79,00,00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardSvr\Security]
"Security"=hex:01,00,14,80,80,00,00,00,8c,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,50,00,03,00,00,00,00,00,14,00,ff,01,0f,00,01,01,00,00,00,00,00,\
  03,00,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,00,14,00,9d,00,02,00,01,01,00,00,00,00,00,01,00,00,00,00,73,\
  00,44,49,52,45,43,54,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,\
  00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardSvr\Enum]
"0"="Root\\LEGACY_SCARDSVR\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Schedule]
"Description"="Enables a program to run at a designated time."
"Type"=dword:00000120
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,4d,53,54,61,73,6b,2e,65,78,65,00
"DisplayName"="Task Scheduler"
"DependOnService"=hex(7):52,70,63,53,73,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"NextAtJobId"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Schedule\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Schedule\Enum]
"0"="Root\\LEGACY_SCHEDULE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SchedulingAgent]
"Start"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\seclogon]
"Type"=dword:00000120
"Start"=dword:00000002
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"DisplayName"="RunAs Service"
"ObjectName"="LocalSystem"
"Description"="Enables starting processes under alternate credentials"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\seclogon\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\seclogon\Enum]
"0"="Root\\LEGACY_SECLOGON\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SENS]
"DependOnService"=hex(7):45,76,65,6e,74,53,79,73,74,65,6d,00,00
"Description"="Tracks system events such as Windows logon, network, and power
events.  Notifies COM+ Event System subscribers of these events."
"DisplayName"="System Event Notification"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"ObjectName"="LocalSystem"
"Group"="Network"
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SENS\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,\
  33,32,5c,73,65,6e,73,2e,64,6c,6c,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SENS\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SENS\Enum]
"0"="Root\\LEGACY_SENS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\serenum]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"DisplayName"="Serenum Filter Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,73,65,72,\
  65,6e,75,6d,2e,73,79,73,00
"Group"="PNP Filter"
"Tag"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\serenum\Enum]
"0"="ACPI\\PNP0501\\0"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="ACPI\\PNP0501\\1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Serial]
"ErrorControl"=dword:00000000
"Group"="Extended base"
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001
"ForceFifoEnable"=dword:00000001
"RxFIFO"=dword:00000008
"TxFIFO"=dword:0000000e
"PermitShare"=dword:00000000
"LogFifo"=dword:00000000
"DisplayName"="Serial port driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,73,65,72,\
  69,61,6c,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Serial\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Serial\Enum]
"0"="ACPI\\PNP0501\\0"
"Count"=dword:00000002
"NextInstance"=dword:00000002
"1"="ACPI\\PNP0501\\1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sfloppy]
"DependOnGroup"=hex(7):53,43,53,49,20,6d,69,6e,69,70,6f,72,74,00,00
"ErrorControl"=dword:00000000
"Group"="Primary disk"
"Start"=dword:00000001
"Tag"=dword:00000004
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sfloppy\Enum]
"Count"=dword:00000000
"NextInstance"=dword:00000000
"INITSTARTFAILED"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sglfb]
"ErrorControl"=dword:00000001
"Group"="Video"
"Start"=dword:00000001
"Type"=dword:00000001
"BootDriver"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sglfb\Device0]
"InstalledDisplayDrivers"=hex(7):73,67,6c,66,62,00,00
"VgaCompatible"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess]
"Type"=dword:00000120
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,6e,65,74,73,76,63,73,00
"DisplayName"="Internet Connection Sharing"
"DependOnService"=hex(7):52,61,73,4d,61,6e,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides network address translation, addressing, and name resolution
services for all computers on your home network through a dial-up connection."

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,69,70,6e,61,74,68,6c,70,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SharedAccess\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Simbad]
"ErrorControl"=dword:00000001
"Group"="Filter"
"Start"=dword:00000004
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sparrow]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:00000007
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sparrow\Parameters]
"LegacyAdapterDetection"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Sparrow\Parameters\PnpInterfac
e]
"1"=dword:00000001
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler]
"DependOnService"=hex(7):52,50,43,53,53,00,00
"Description"="Loads files to memory for later printing."
"DisplayName"="Print Spooler"
"ErrorControl"=dword:00000001
"Group"="SpoolerGroup"
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,70,6f,6f,6c,73,76,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000110

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler\Performance]
"Close"="PerfClose"
"Collect"="PerfCollect"
"Collect Timeout"=dword:000007d0
"Library"="winspool.drv"
"Object List"="1450"
"Open"="PerfOpen"
"Open Timeout"=dword:00000fa0
"WbemAdapFileTime"=hex:00,68,a1,1d,a4,a5,bf,01
"WbemAdapFileSize"=dword:0001b710
"WbemAdapStatus"=dword:ffffffff

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Spooler\Enum]
"0"="Root\\LEGACY_SPOOLER\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLSERVERAGENT]
"Type"=dword:00000010
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,3a,5c,50,52,4f,47,52,41,7e,31,5c,4d,49,43,52,4f,53,7e,32,\
  5c,4d,53,53,51,4c,5c,62,69,6e,6e,5c,73,71,6c,61,67,65,6e,74,2e,65,78,65,00
"DisplayName"="SQLSERVERAGENT"
"DependOnService"=hex(7):4d,53,53,51,4c,53,45,52,56,45,52,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SQLSERVERAGENT\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,30,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,5c,00,44,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,5c,00,44,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Srv]
"Type"=dword:00000002
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"Tag"=dword:00000006
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,73,72,76,\
  2e,73,79,73,00
"DisplayName"="Srv"
"Group"="Network"
"Description"="Srv"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Srv\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Srv\Enum]
"0"="Root\\LEGACY_SRV\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000001
"DisplayName"="Software Bus Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,73,77,65,\
  6e,75,6d,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{96e080c7-143c-
11d1-b40f-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{96e080c7-143c-
11d1-b40f-00a0c9223196}\{3C0D501A-140B-11D1-B40F-00A0C9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{96e080c7-143c-
11d1-b40f-00a0c9223196}\{3C0D501A-140B-11D1-B40F-00A0C9223196}\{3c0d501a-140b-11d1-
b40f-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{97ebaacc-95bd-
11d0-a3ea-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{97ebaacc-95bd-
11d0-a3ea-00a0c9223196}\{53172480-4791-11D0-A5D6-28DB04C10000}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{97ebaacc-95bd-
11d0-a3ea-00a0c9223196}\{53172480-4791-11D0-A5D6-28DB04C10000}\{53172480-4791-11d0-
a5d6-28db04c10000}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{085AFF00-62CE-11CF-A5D6-28DB04C10000}]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{085AFF00-62CE-11CF-A5D6-28DB04C10000}\{085aff00-62ce-11cf-
a5d6-28db04c10000}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{65E8773D-8F56-11D0-A3B9-00A0C9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{65E8773D-8F56-11D0-A3B9-00A0C9223196}\{65e8773d-8f56-11d0-
a3b9-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{65E8773E-8F56-11D0-A3B9-00A0C9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{c68127b1-9bea-
11d0-8fa5-00c04fc324c1}\{65E8773E-8F56-11D0-A3B9-00A0C9223196}\{65e8773e-8f56-11d0-
a3b9-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{ddf4358e-bb2c-
11d0-a42f-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{ddf4358e-bb2c-
11d0-a42f-00a0c9223196}\{97EBAACB-95BD-11D0-A3EA-00A0C9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{ddf4358e-bb2c-
11d0-a42f-00a0c9223196}\{97EBAACB-95BD-11D0-A3EA-00A0C9223196}\{97ebaacb-95bd-11d0-
a3ea-00a0c9223196}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{eeab7790-c514-
11d1-b42b-00805fc1270e}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{eeab7790-c514-
11d1-b42b-00805fc1270e}\asyncmac]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Devices\{eeab7790-c514-
11d1-b42b-00805fc1270e}\asyncmac\{ad498944-762f-11d0-8dcb-00c04fc3358c}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\swenum\Enum]
"0"="Root\\SYSTEM\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc810]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000001a
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc810\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc810\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc8xx]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Type"=dword:00000001
"Tag"=dword:00000036

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc8xx\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\symc8xx\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sym_hi]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Type"=dword:00000001
"Tag"=dword:00000037

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sym_hi\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\sym_hi\Parameters\PnpInterface
]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SysmonLog]
"Description"="Configures performance logs and alerts."
"DisplayName"="Performance Logs and Alerts"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,6d,6c,6f,67,73,76,63,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000010
"DefaultLogFileFolder"=hex(2):25,53,79,73,74,65,6d,44,72,69,76,65,25,5c,50,65,\
  72,66,4c,6f,67,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TapiSrv]
"DependOnService"=hex(7):50,6c,75,67,50,6c,61,79,00,52,70,63,53,73,00,00
"Description"="Provides Telephony API (TAPI) support for programs that control
telephony devices and IP based voice connections on the local computer and, through
the LAN, on servers that are also running the service."
"DisplayName"="Telephony"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,76,63,68,6f,73,74,2e,65,78,65,20,2d,6b,20,74,61,70,69,73,72,76,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000120

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TapiSrv\Parameters]
"ServiceDll"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,74,61,70,69,73,72,76,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TapiSrv\Performance]
"Close"="CloseTapiPerformanceData"
"Collect"="CollectTapiPerformanceData"
"Library"="tapiperf.dll"
"ObjectList"="1150"
"Open"="OpenTapiPerformanceData"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00001910
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TapiSrv\Security]
"Security"=hex:01,00,14,80,ac,00,00,00,b8,00,00,00,14,00,00,00,34,00,00,00,02,\
  00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,20,02,00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,\
  00,00,00,01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,\
```

```
  00,00,00,23,02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,\
  00,00,00,03,18,00,9d,00,00,00,01,02,00,00,00,00,00,05,20,00,00,00,21,02,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TapiSrv\Enum]
"0"="Root\\LEGACY_TAPISRV\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip]
"Type"=dword:00000001
"Start"=dword:00000001
"ErrorControl"=dword:00000001
"Tag"=dword:00000004
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,74,63,70,\
  69,70,2e,73,79,73,00
"DisplayName"="TCP/IP Protocol Driver"
"Group"="PNP_TDI"
"Description"="TCP/IP Protocol Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Linkage]
"Bind"=hex(7):5c,44,65,76,69,63,65,5c,7b,32,36,32,41,34,45,31,32,2d,44,33,39,\
  42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,\
  00,5c,44,65,76,69,63,65,5c,4e,64,69,73,57,61,6e,49,70,00,00
"Route"=hex(7):22,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,42,31,2d,\
  41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,22,00,22,4e,64,69,73,\
  57,61,6e,49,70,22,00,00
"Export"=hex(7):5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,32,36,32,41,34,45,\
  31,32,2d,44,33,39,42,2d,34,36,42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,\
  41,41,35,31,44,7d,00,5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,32,31,43,\
  44,35,35,37,34,2d,31,42,41,35,2d,34,39,36,38,2d,41,31,35,36,2d,45,42,30,46,\
  38,35,36,43,43,38,34,44,7d,00,5c,44,65,76,69,63,65,5c,54,63,70,69,70,5f,7b,\
  30,44,39,32,42,34,31,42,2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,39,\
  34,46,44,36,39,46,42,32,45,34,41,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters]
"NV Hostname"="cheapo"
"DataBasePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,64,72,69,76,65,72,73,5c,65,74,63,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="cheapo"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\Ndis
WanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,63,70,69,70,5c,50,61,72,61,6d,65,74,65,72,73,5c,49,6e,74,\
  65,72,66,61,63,65,73,5c,7b,32,31,43,44,35,35,37,34,2d,31,42,41,35,2d,34,39,\
  36,38,2d,41,31,35,36,2d,45,42,30,46,38,35,36,43,43,38,34,44,7d,00,54,63,70,\
  69,70,5c,50,61,72,61,6d,65,74,65,72,73,5c,49,6e,74,65,72,66,61,63,65,73,5c,\
  7b,30,44,39,32,42,34,31,42,2d,37,32,39,31,2d,34,34,33,42,2d,42,36,46,31,2d,\
  39,34,46,44,36,39,46,42,32,45,34,41,7d,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:74,55,cd,21,a5,1b,68,49,a1,56,eb,0f,85,6c,c8,4d,1b,b4,92,0d,\
  91,72,3b,44,b6,f1,94,fd,69,fb,2e,4a

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters\{262
A4E12-D39B-46B1-AEA7-F1B0EE8AA51D}]
"LLInterface"=""
"IpConfig"=hex(7):54,63,70,69,70,5c,50,61,72,61,6d,65,74,65,72,73,5c,49,6e,74,\
  65,72,66,61,63,65,73,5c,7b,32,36,32,41,34,45,31,32,2d,44,33,39,42,2d,34,36,\
  42,31,2d,41,45,41,37,2d,46,31,42,30,45,45,38,41,41,35,31,44,7d,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\DNSRegistered
Adapters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{0
D92B41B-7291-443B-B6F1-94FD69FB2E4A}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,2e,30,2e,30,2e,30,00,00
"SubnetMask"=hex(7):30,2e,30,2e,30,2e,30,00,00
"DefaultGateway"=hex(7):00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{2
1CD5574-1BA5-4968-A156-EB0F856CC84D}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,2e,30,2e,30,2e,30,00,00
"SubnetMask"=hex(7):30,2e,30,2e,30,2e,30,00,00
"DefaultGateway"=hex(7):00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{2
62A4E12-D39B-46B1-AEA7-F1B0EE8AA51D}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,33,30,2e,31,36,38,2e,32,30,36,2e,39,39,00,00
"SubnetMask"=hex(7):32,35,35,2e,32,35,35,2e,30,2e,30,00,00
"DefaultGateway"=hex(7):00
"DefaultGatewayMetric"=hex(7):00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00
"UDPAllowedPorts"=hex(7):30,00,00
"RawIPAllowedProtocols"=hex(7):30,00,00
"NTEContextList"=hex(7):30,78,30,30,30,30,30,30,30,32,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:398ed4c1
"T1"=dword:398edbc9
"T2"=dword:398ee10f
"LeaseTerminatesTime"=dword:398ee2d1
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
```

```
"IPAutoconfigurationSeed"=dword:00000000
"AddressType"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\PersistentRou
tes]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Winsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,\
  6d,33,32,5c,77,73,68,74,63,70,69,70,2e,64,6c,6c,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,00,00,00,01,00,00,00,06,00,00,00,02,\
  00,00,00,01,00,00,00,00,00,00,00,02,00,00,00,00,00,00,00,06,00,00,00,00,00,\
  00,00,00,00,00,00,06,00,00,00,00,00,00,00,01,00,00,00,06,00,00,00,02,00,00,\
  00,02,00,00,00,11,00,00,00,02,00,00,00,02,00,00,00,00,00,00,00,02,00,00,00,\
  00,00,00,00,11,00,00,00,00,00,00,00,00,00,00,11,00,00,00,00,00,00,00,02,\
  00,00,00,11,00,00,00,02,00,00,00,03,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Performance]
"Close"="CloseTcpIpPerformanceData"
"Collect"="CollectTcpIpPerformanceData"
"Library"="Perfctrs.dll"
"Open"="OpenTcpIpPerformanceData"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:0000a310
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,02,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,03,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,03,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\ServiceProvider]
"Class"=dword:00000008
"DnsPriority"=dword:000007d0
"HostsPriority"=dword:000001f4
"LocalPriority"=dword:000001f3
"ProviderPath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,\
  33,32,5c,77,73,6f,63,6b,33,32,2e,64,6c,6c,00
"NetbtPriority"=dword:000007d1
"Name"="TCP/IP"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Enum]
"0"="Root\\LEGACY_TCPIP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDASYNC]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDIPX]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDNETB]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDPIPE]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDSPX]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDTCP]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TDTCP\Enum]
"0"="Root\\LEGACY_TDTCP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermDD]
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,74,65,72,6d,64,64,2e,73,79,73,00
"Type"=dword:00000001
"DisplayName"="Terminal Device Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermDD\Enum]
"0"="Root\\LEGACY_TERMDD\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService]
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,74,65,72,6d,73,72,76,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Type"=dword:00000010
"Description"="Provides a multisession environment that allows client devices to
access a virtual Windows 2000 Professional desktop session and Windows-based
programs running on the server."
"DisplayName"="Terminal Services"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Parameters]
"UseLicenseServer"=dword:00000000
"fInternetConnector"=dword:00000000
"X509 Certificate ID"=hex:02,00,00,00,0d,06,ba,94,0c,66,d1,30,5e,4f,de,16,e3,\
  17,84,d3
"X509 Certificate"=hex:02,00,00,80,02,00,00,00,63,01,00,00,30,82,01,5f,30,82,\
  01,0d,a0,03,02,01,02,02,08,01,9e,58,f5,ae,d3,59,90,30,09,06,05,2b,0e,03,02,\
  1d,05,00,30,2e,31,2c,30,0f,06,03,55,04,03,1e,08,00,46,00,45,00,34,00,30,30,\
  19,06,03,55,04,07,1e,12,00,44,00,42,00,5f,00,41,00,50,00,50,00,53,00,52,00,\
  56,30,1e,17,0d,37,30,30,38,30,31,31,35,31,34,31,38,5a,17,0d,34,39,30,38,30,\
  31,31,35,31,34,31,38,5a,30,2e,31,2c,30,0f,06,03,55,04,03,1e,08,00,46,00,45,\
  00,34,00,30,30,19,06,03,55,04,07,1e,12,00,44,00,42,00,5f,00,41,00,50,00,50,\
  00,53,00,52,00,56,30,5c,30,0d,06,09,2a,86,48,86,f7,0d,01,01,01,05,00,03,4b,\
```

```
  00,30,48,02,41,00,e3,cd,db,57,bd,46,27,97,69,bf,84,9d,83,c7,d1,7a,80,3a,19,\
  85,8b,49,a3,23,a3,2b,3b,de,eb,65,0a,d6,08,79,09,86,6f,e8,f8,88,82,25,85,6c,\
  26,9a,b9,90,9a,7a,ac,79,11,59,85,bb,7c,3c,78,9e,c7,fd,88,fb,02,03,01,00,01,\
  a3,13,30,11,30,0f,06,03,55,1d,13,04,08,30,06,01,01,ff,02,01,00,30,09,06,05,\
  2b,0e,03,02,1d,05,00,03,41,00,d5,33,ce,8a,a9,2c,68,32,38,7b,e9,1f,49,47,62,\
  82,6e,f1,f7,e5,fc,d2,c3,71,15,4d,64,cd,a0,2a,85,91,ca,dd,c9,ca,a9,d4,d3,e4,\
  b2,9d,f8,80,3f,c0,82,96,38,2f,47,00,d9,52,85,c9,5e,34,9c,62,2c,22,a0,7c,6f,\
  03,00,00,30,82,03,6b,30,82,03,19,a0,03,02,01,02,02,05,01,00,00,00,02,30,09,\
  06,05,2b,0e,03,02,1d,05,00,30,2e,31,2c,30,0f,06,03,55,04,03,1e,08,00,46,00,\
  45,00,34,00,30,30,19,06,03,55,04,07,1e,12,00,44,00,42,00,5f,00,41,00,50,00,\
  50,00,53,00,52,00,56,30,1e,17,0d,38,30,30,31,30,31,30,36,30,30,30,30,5a,17,\
  0d,33,38,30,31,31,39,30,33,31,34,30,37,5a,30,81,96,31,81,93,30,25,06,03,55,\
  04,03,1e,1e,00,6e,00,63,00,61,00,63,00,6e,00,5f,00,6e,00,70,00,3a,00,43,00,\
  48,00,45,00,41,00,50,00,4f,30,25,06,03,55,04,07,1e,1e,00,6e,00,63,00,61,00,\
  63,00,6e,00,5f,00,6e,00,70,00,3a,00,43,00,48,00,45,00,41,00,50,00,4f,30,43,\
  06,03,55,04,05,1e,3c,00,31,00,42,00,63,00,4b,00,65,00,53,00,44,00,70,00,72,\
  00,78,00,6c,00,39,00,51,00,6f,00,6b,00,63,00,56,00,73,00,42,00,6d,00,2b,00,\
  46,00,30,00,69,00,2f,00,56,00,34,00,3d,00,0d,00,0a,30,5c,30,0d,06,09,2a,86,\
  48,86,f7,0d,01,01,04,05,00,03,4b,00,30,48,02,41,00,c7,fa,9d,e8,c1,cd,22,db,\
  82,9e,f1,28,60,08,0d,fb,6d,64,71,9c,e9,c7,69,7a,9c,9e,7e,ca,b9,89,c3,53,78,\
  30,4a,ed,7b,80,79,5a,01,33,6c,28,cc,95,fd,3e,26,2f,23,7f,48,ca,f1,15,6e,04,\
  2c,d2,51,ae,05,c5,02,03,01,00,01,a3,82,01,b7,30,82,01,b3,30,14,06,09,2b,06,\
  01,04,01,82,37,12,04,01,01,ff,04,04,01,00,05,00,30,3c,06,09,2b,06,01,04,01,\
  82,37,12,02,01,01,ff,04,2c,4d,00,69,00,63,00,72,00,6f,00,73,00,6f,00,66,00,\
  74,00,20,00,43,00,6f,00,72,00,70,00,6f,00,72,00,61,00,74,00,69,00,6f,00,6e,\
  00,00,00,30,81,cd,06,09,2b,06,01,04,01,82,37,12,05,01,01,ff,04,81,bc,00,30,\
  00,00,01,00,00,00,02,00,00,00,09,04,00,00,1c,00,4a,00,66,00,4a,00,b0,00,01,\
  00,33,00,64,00,32,00,36,00,37,00,39,00,35,00,34,00,2d,00,65,00,65,00,62,00,\
  37,00,2d,00,31,00,31,00,64,00,31,00,2d,00,62,00,39,00,34,00,65,00,2d,00,30,\
  00,30,00,63,00,30,00,34,00,66,00,61,00,33,00,30,00,38,00,30,00,64,00,00,00,\
  33,00,64,00,32,00,36,00,37,00,39,00,35,00,34,00,2d,00,65,00,65,00,62,00,37,\
  00,2d,00,31,00,31,00,64,00,31,00,2d,00,62,00,39,00,34,00,65,00,2d,00,30,00,\
  30,00,63,00,30,00,34,00,66,00,61,00,33,00,30,00,38,00,30,00,64,00,00,00,00,\
  00,00,10,00,80,d4,00,00,00,00,00,00,30,6a,06,09,2b,06,01,04,01,82,37,12,06,01,\
  01,ff,04,5a,00,30,00,00,00,00,0a,00,3a,00,46,00,45,00,34,00,30,00,00,00,35,\
  00,31,00,38,00,37,00,36,00,2d,00,4f,00,45,00,4d,00,2d,00,30,00,30,00,30,00,\
  30,00,30,00,30,00,37,00,2d,00,35,00,39,00,31,00,35,00,32,00,00,00,44,00,42,\
  00,5f,00,41,00,50,00,50,00,53,00,52,00,56,00,00,00,00,00,30,21,06,03,55,1d,\
  23,01,01,ff,04,17,30,15,a1,0c,a4,0a,46,00,45,00,34,00,30,00,00,00,82,05,01,\
  00,00,00,02,30,09,06,05,2b,0e,03,02,1d,05,00,03,41,00,be,4f,a7,08,1f,2a,3e,\
  a0,b7,6f,48,12,1c,a1,9b,89,9a,94,c4,3b,c3,04,e5,e1,ed,b7,3d,da,95,97,a0,93,\
  2b,6c,5e,85,50,e9,2f,85,66,36,19,8f,5b,02,1f,2a,c5,b0,b9,a5,1a,4f,92,ff,20,\
  5c,ea,41,8f,47,18,7a,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
"LicensingGracePeriodExpirationWarningDays"=dword:0000000f

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Performance]
"Close"="CloseTSObject"
"Collect Timeout"=dword:000003e8
"Collect"="CollectTSObjectData"
"Open Timeout"=dword:000003e8
"Open"="OpenTSObject"
"Library"="perfts.dll"
"Last Counter"=dword:000008d8
"Last Help"=dword:000008d9
"First Counter"=dword:00000858
"First Help"=dword:00000859
"Object List"="2136 2258"
"WbemAdapFileTime"=hex:00,60,4e,96,aa,40,bf,01
"WbemAdapFileSize"=dword:00003110
"WbemAdapStatus"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService\Enum]
"0"="Root\\LEGACY_TERMSERVICE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tga]
"BootDriver"=dword:00000001
"ErrorControl"=dword:00000000
"Group"="Video"
"Start"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\tga\Device0]
"InstalledDisplayDrivers"=hex(7):74,67,61,00,00
"VgaCompatible"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TlntSvr]
"DependOnService"=hex(7):52,70,63,53,73,00,54,63,70,49,70,00,00
"Description"="Allows a remote user to log on to the system and run console programs
using the command line."
"DisplayName"="Telnet"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,74,6c,6e,74,73,76,72,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkSvr]
"Description"="Stores information so that files moved between volumes can be tracked
for each volume in the domain."
"DisplayName"="Distributed Link Tracking Server"
"DependOnService"=hex(7):52,70,63,53,73,00,00
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkSvr\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkWks]
"Description"="Sends notifications of files moving between NTFS volumes in a network
domain."
"DisplayName"="Distributed Link Tracking Client"
"DependOnService"=hex(7):52,70,63,53,73,00,00
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000002
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkWks\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkWks\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,18,00,9d,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  23,02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,20,\
  02,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,\
  00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,\
  00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TrkWks\Enum]
"0"="Root\\LEGACY_TRKWKS\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Udfs]
"ErrorControl"=dword:00000001
"Group"="File system"
"Start"=dword:00000004
"Type"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ultra66]
"ErrorControl"=dword:00000001
"Group"="SCSI miniport"
"Start"=dword:00000004
"Tag"=dword:0000003b
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ultra66\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ultra66\Parameters\PnpInterfac
e]
"5"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Update]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000001
"DisplayName"="Microcode Update Driver"
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,75,70,64,\
  61,74,65,2e,73,79,73,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Update\Devices]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Update\Enum]
"0"="Root\\SYSTEM\\0001"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\UPS]
"Description"="Manages an uninterruptible power supply (UPS) connected to the
computer."
"DisplayName"="Uninterruptible Power Supply"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,75,70,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000010

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\UtilMan]
"Description"="Starts and configures accessibility tools from one window "
"DisplayName"="Utility Manager"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,55,74,69,6c,4d,61,6e,2e,65,78,65,00
"Start"=dword:00000003
"ObjectName"="LocalSystem"
"Type"=dword:00000110

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\UtilMan\Security]
"Security"=hex:01,00,14,80,bc,00,00,00,c8,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,8c,00,05,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,00,00,\
  05,0b,00,00,00,00,00,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,23,02,00,00,9d,37,5c,92,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,\
  00,00,05,20,00,00,00,20,02,00,00,9d,37,5c,92,00,00,1c,00,ff,01,0f,00,01,02,\
  00,00,00,00,00,05,20,00,00,00,25,02,00,00,9d,37,5c,92,00,00,18,00,fd,01,02,\
  00,01,01,00,00,00,00,00,05,12,00,00,00,25,02,00,00,01,01,00,00,00,00,00,05,\
  12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VgaSave]
"ErrorControl"=dword:00000000
"Group"="Video Save"
"ImagePath"=hex(2):5c,53,79,73,74,65,6d,52,6f,6f,74,5c,53,79,73,74,65,6d,33,32,\
  5c,64,72,69,76,65,72,73,5c,76,67,61,2e,73,79,73,00
"Start"=dword:00000001
"Tag"=dword:00000001
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VgaSave\Device0]
"InstalledDisplayDrivers"=hex(7):76,67,61,00,00
"VgaCompatible"=dword:00000001
"HardwareInformation.Crc32"=hex:76,a0,56,21

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VgaSave\Enum]
"0"="Root\\LEGACY_VGASAVE\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time]
"Description"="Sets the computer clock."
"DisplayName"="Windows Time"
"ErrorControl"=dword:00000001
"Group"=""
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,73,65,72,76,69,63,65,73,2e,65,78,65,00
"Objectname"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Parameters]
"LocalNTP"=dword:00000000
"Period"="SpecialSkew"
"type"="Nt5DS"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Security]
"Security"=hex:01,00,14,80,a8,00,00,00,b4,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,78,00,05,00,00,00,00,03,14,00,8d,00,02,00,01,01,00,00,00,00,00,\
  01,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\
  20,02,00,00,00,03,18,00,8d,00,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,\
  02,00,00,00,03,14,00,9d,00,00,00,01,01,00,00,00,00,00,05,04,00,00,00,00,03,\
  18,00,9d,00,00,00,01,02,00,00,00,00,00,05,20,00,00,00,21,02,00,00,01,01,00,\
  00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\Adv
ancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDS
Server.DataFactory]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual
Roots]
"/MSADC"=hex(2):5c,73,79,73,74,65,6d,5c,6d,73,61,64,63,2c,2c,35,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wanarp]
"Type"=dword:00000001
"Start"=dword:00000003
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,77,61,6e,\
  61,72,70,2e,73,79,73,00
"DisplayName"="Remote Access IP ARP Driver"
"Description"="Remote Access IP ARP Driver"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wanarp\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,61,00,6c,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,6c,00,65,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,6c,00,65,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wanarp\Enum]
"0"="Root\\LEGACY_WANARP\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WDICA]
"ErrorControl"=dword:00000000
"Start"=dword:00000003
"Type"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinMgmt]
"Type"=dword:00000010
"Start"=dword:00000002
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,65,6d,33,\
  32,5c,57,42,45,4d,5c,57,69,6e,4d,67,6d,74,2e,65,78,65,00
"DisplayName"="Windows Management Instrumentation"
"DependOnService"=hex(7):52,50,43,53,53,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"
"Description"="Provides system management information."
"FailureActions"=hex:80,51,01,00,00,00,00,00,00,00,00,00,02,00,00,00,10,df,08,\
  00,01,00,00,00,60,ea,00,00,01,00,00,00,60,ea,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinMgmt\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,20,02,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinMgmt\Enum]
"0"="Root\\LEGACY_WINMGMT\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock]
"ErrorControl"=dword:00000001
"Start"=dword:00000003
"Type"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Parameters]
"Transports"=hex(7):54,63,70,69,70,00,4e,65,74,42,49,4f,53,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Setup Migration]
"Setup Version"=dword:00001009
"Provider List"=hex(7):4e,65,74,42,49,4f,53,00,54,63,70,69,70,00,00
"Known Static Providers"=hex(7):54,63,70,69,70,00,4e,77,6c,6e,6b,49,70,78,00,\
  4e,77,6c,6e,6b,53,70,78,00,41,70,70,6c,65,54,61,6c,6b,00,49,73,6f,54,70,00,\
  00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Setup
Migration\Providers]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Setup
Migration\Providers\NetBIOS]
"WinSock 1.1 Provider Data"=hex:0e,10,00,00,11,00,00,00,14,00,00,00,14,00,00,\
  00,05,00,00,00,00,00,00,00,80,00,fa,00,00,0e,03,00,00,09,12,00,00,11,00,00,00,\
  14,00,00,00,14,00,00,00,02,00,00,00,00,00,00,80,00,fa,00,00,98,02,00,00,0e,\
  10,00,00,11,00,00,00,14,00,00,00,14,00,00,00,05,00,00,00,ff,ff,ff,ff,00,fa,\
  00,00,22,02,00,00,09,12,00,00,11,00,00,00,14,00,00,00,14,00,00,00,02,00,00,\
  00,ff,ff,ff,ff,00,fa,00,00,ac,01,00,00,0e,10,00,00,11,00,00,00,14,00,00,00,\
  14,00,00,00,05,00,00,00,fe,ff,ff,ff,00,fa,00,00,36,01,00,00,09,12,00,00,11,\
  00,00,00,14,00,00,00,14,00,00,00,02,00,00,00,fe,ff,ff,ff,00,fa,00,00,c0,00,\
  00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,4e,00,65,00,74,00,42,\
  00,54,00,5f,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,30,00,44,00,39,00,\
  32,00,42,00,34,00,31,00,42,00,2d,00,37,00,32,00,39,00,31,00,2d,00,34,00,34,\
  00,33,00,42,00,2d,00,42,00,36,00,46,00,31,00,2d,00,39,00,34,00,46,00,44,00,\
  36,00,39,00,46,00,42,00,32,00,45,00,34,00,41,00,7d,00,00,00,5c,00,44,00,65,\
  00,76,00,69,00,63,00,65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,\
  63,00,70,00,69,00,70,00,5f,00,7b,00,30,00,44,00,39,00,32,00,42,00,34,00,31,\
  00,42,00,2d,00,37,00,32,00,39,00,31,00,2d,00,34,00,34,00,33,00,42,00,2d,00,\
  42,00,36,00,46,00,31,00,2d,00,39,00,34,00,46,00,44,00,36,00,39,00,46,00,42,\
  00,32,00,45,00,34,00,41,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,32,00,31,00,43,00,44,00,35,00,35,00,37,00,34,00,2d,00,31,00,\
  42,00,41,00,35,00,2d,00,34,00,39,00,36,00,38,00,2d,00,41,00,31,00,35,00,36,\
  00,2d,00,45,00,42,00,30,00,46,00,38,00,35,00,36,00,43,00,43,00,38,00,34,00,\
  44,00,7d,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,4e,00,65,\
  00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,32,00,\
  31,00,43,00,44,00,35,00,35,00,37,00,34,00,2d,00,31,00,42,00,41,00,35,00,2d,\
  00,34,00,39,00,36,00,38,00,2d,00,41,00,31,00,35,00,36,00,2d,00,45,00,42,00,\
  30,00,46,00,38,00,35,00,36,00,43,00,43,00,38,00,34,00,44,00,7d,00,00,00,5c,\
  00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,\
  5f,00,54,00,63,00,70,00,69,00,70,00,5f,00,7b,00,32,00,36,00,32,00,41,00,34,\
  00,45,00,31,00,32,00,2d,00,44,00,33,00,39,00,42,00,2d,00,34,00,36,00,42,00,\
  31,00,2d,00,41,00,45,00,41,00,37,00,2d,00,46,00,31,00,42,00,30,00,45,00,45,\
  00,38,00,41,00,41,00,35,00,31,00,44,00,7d,00,00,00,5c,00,44,00,65,00,76,00,\
  69,00,63,00,65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,\
  00,69,00,70,00,5f,00,7b,00,32,00,36,00,32,00,41,00,34,00,45,00,31,00,32,00,\
  2d,00,44,00,33,00,39,00,42,00,2d,00,34,00,36,00,42,00,31,00,2d,00,41,00,45,\
  00,41,00,37,00,2d,00,46,00,31,00,42,00,30,00,45,00,45,00,38,00,41,00,41,00,\
  35,00,31,00,44,00,7d,00,00,00
"WinSock 2.0 Provider ID"=hex:30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,48,a1,92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Setup
Migration\Providers\Tcpip]
```

```
"WinSock 2.0 Provider ID"=hex:a0,1a,0f,e7,8b,ab,cf,11,8c,a3,00,80,5f,48,a1,92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Winsock\Setup Migration\Well
Known Guids]
"IsoTp"=hex:b0,cb,e4,89,c1,b9,cf,11,95,c8,00,80,5f,48,a1,92
"McsXns"=hex:b1,cb,e4,89,c1,b9,cf,11,95,c8,00,80,5f,48,a1,92
"AppleTalk"=hex:a0,17,3b,2c,df,c6,cf,11,95,c8,00,80,5f,48,a1,92

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters]
"WinSock_Registry_Version"="2.0"
"Current_NameSpace_Catalog"="NameSpace_Catalog5"
"Current_Protocol_Catalog"="Protocol_Catalog9"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_
Catalog5]
"Num_Catalog_Entries"=dword:00000002
"Serial_Access_Num"=dword:00000003

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_
Catalog5\Catalog_Entries]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_
Catalog5\Catalog_Entries\000000000001]
"LibraryPath"="%SystemRoot%\\System32\\rnr20.dll"
"DisplayString"="Tcpip"
"ProviderId"=hex:40,9d,05,22,9e,7e,cf,11,ae,5a,00,aa,00,a7,11,2b
"SupportedNameSpace"=dword:0000000c
"Enabled"=dword:00000001
"Version"=dword:00000000
"StoresServiceClassInfo"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\NameSpace_
Catalog5\Catalog_Entries\000000000002]
"LibraryPath"="%SystemRoot%\\System32\\winrnr.dll"
"DisplayString"="NTDS"
"ProviderId"=hex:ee,37,26,3b,80,e5,cf,11,a5,55,00,c0,4f,d8,d4,ac
"SupportedNameSpace"=dword:00000020
"Enabled"=dword:00000001
"Version"=dword:00000000
"StoresServiceClassInfo"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9]
"Num_Catalog_Entries"=dword:0000000b
"Next_Catalog_Entry_ID"=dword:000003f4
"Serial_Access_Num"=dword:00000004

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000001]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,66,00,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,a0,1a,0f,e7,8b,ab,cf,11,8c,a3,00,80,5f,\
  48,a1,92,e9,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,02,00,00,00,10,00,00,\
  00,10,00,00,00,01,00,00,00,06,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,54,00,63,00,70,\
  00,69,00,70,00,20,00,5b,00,54,00,43,00,50,00,2f,00,49,00,50,00,5d,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000002]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,06,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,a0,1a,0f,e7,8b,ab,cf,11,8c,a3,00,80,5f,\
  48,a1,92,ea,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,02,00,00,00,10,00,00,\
  00,10,00,00,00,02,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  bb,ff,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,54,00,63,00,70,\
  00,69,00,70,00,20,00,5b,00,55,00,44,00,50,00,2f,00,49,00,50,00,5d,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
```

```
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000003]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,06,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,0c,00,00,00,a0,1a,0f,e7,8b,ab,cf,11,8c,a3,00,80,5f,\
  48,a1,92,eb,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,02,00,00,00,10,00,00,\
  00,10,00,00,00,03,00,00,00,00,00,00,ff,00,00,00,00,00,00,00,00,00,00,00,00,\
  bb,ff,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,54,00,63,00,70,\
  00,69,00,70,00,20,00,5b,00,52,00,41,00,57,00,2f,00,49,00,50,00,5d,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000004]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,72,73,76,70,73,70,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,26,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,e0,a9,60,9d,7a,33,d0,11,bd,88,00,00,c0,\
  82,e6,9a,ec,03,00,00,01,00,00,00,2c,3c,4b,6a,00,00,00,00,e4,f6,d2,06,c6,82,\
  4c,6a,00,00,00,00,2c,3c,4b,6a,00,01,00,00,04,00,00,00,02,00,00,00,10,00,00,\
  00,10,00,00,00,02,00,00,00,11,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  bb,ff,00,00,00,00,00,00,52,00,53,00,56,00,50,00,20,00,55,00,44,00,50,00,20,\
  00,53,00,65,00,72,00,76,00,69,00,63,00,65,00,20,00,50,00,72,00,6f,00,76,00,\
  69,00,64,00,65,00,72,00,00,00,02,00,00,00,6e,b5,f8,77,34,00,00,c0,04,f8,d2,\
  06,07,7f,db,77,10,f8,d2,06,00,00,00,00,64,f7,d2,06,00,00,00,00,6e,b5,f8,77,\
  00,00,00,00,e8,f7,d2,06,70,7c,db,77,00,00,00,00,44,3c,4b,6a,ec,08,00,00,ec,\
  08,00,00,00,00,00,00,50,9a,fc,77,67,9a,fc,77,c8,48,34,05,6c,fa,d2,06,c8,48,\
  34,05,00,00,00,00,5c,00,44,00,65,00,76,00,69,00,63,00,65,00,5c,00,7b,00,32,\
  00,36,00,32,00,41,00,34,00,45,00,31,00,32,00,2d,00,44,00,33,00,39,00,42,00,\
  2d,00,34,00,36,00,42,00,31,00,2d,00,41,00,45,00,41,00,37,00,2d,00,46,00,31,\
  00,42,00,30,00,45,00,45,00,38,00,41,00,41,00,35,00,31,00,44,00,7d,00,00,00,\
  42,00,30,00,45,00,45,00,38,00,41,00,41,00,35,00,31,00,44,00,7d,00,00,00,00,\
  00,02,00,07,80,34,f8,d2,06,96,80,4c,6a,44,09,00,00,2c,3c,4b,6a,3c,f8,d2,06,\
  17,7d,db,77,ec,08,00,00,00,00,00,00,c8,fa,d2,06,dd,7c,db,77,4c,f8,d2,06,a6,\
  7d,db,77,4c,f8,d2,06,7c,6a,4d,6a,00,00,00,00,28,fb,d2,06,44,36,f1,01,08,36,\
  f1,01,8b,19,db,77,c8,7c,db,77,ff,ff,ff,ff,9c,f8,d2,06,38,82,db,77,18,00,00,\
  00,90,f8,d2,06,a0,0f,00,00,50,9a,fc,77,67,9a,fc,77,0e,00,07,80,e8,5d,ec,01,\
  08,36,f1,01,72,00,74,00,d8,f8,d2,06,00,00,00,00,e4,fa,d2,06,a1,5c,4c,6a,02,\
  00,00,80,d8,f8,d2,06,00,00,00,00,19,00,02,00,e0,fa,d2,06,07,21,00,78,1b,b1,\
  4d,6a,b8,a8,16,00,88,01,07,00,2a,b1,4d,6a,00,00,00,00,44,36,f1,01,28,fb,d2,\
  06,53,00,59,00,53,00,54,00,45,00,4d,00,5c,00,43,00,75,00,72,00,72,00,65,00,\
  6e,00,74,00,43,00,6f,00,6e,00,74,00,01,00,00,00,6c,fa,d2,06

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000005]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,72,73,76,70,73,70,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,66,20,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,e0,a9,60,9d,7a,33,d0,11,bd,88,00,00,c0,\
  82,e6,9a,ed,03,00,00,01,00,00,00,6b,00,64,00,a0,0f,00,00,50,9a,fc,77,67,9a,\
  fc,77,01,00,00,00,e4,fb,d2,06,e4,fb,d2,06,04,00,00,00,02,00,00,00,10,00,00,\
  00,10,00,00,00,01,00,00,00,06,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,52,00,53,00,56,00,50,00,20,00,54,00,43,00,50,00,20,\
  00,53,00,65,00,72,00,76,00,69,00,63,00,65,00,20,00,50,00,72,00,6f,00,76,00,\
  69,00,64,00,65,00,72,00,00,00,a0,0f,00,00,50,9a,fc,77,67,9a,fc,77,a4,7e,e5,\
  01,1c,fc,d2,06,08,36,f1,01,15,46,4d,6a,00,00,00,00,88,01,07,00,b8,a8,16,00,\
  d8,48,34,05,a0,fa,d2,06,80,5d,ec,01,54,fa,d2,06,a0,0f,00,00,50,9a,fc,77,67,\
  9a,fc,77,0e,00,07,80,d0,fb,d2,06,08,36,f1,01,90,01,07,00,88,f9,d2,06,6c,fa,\
  d2,06,1c,fc,d2,06,db,80,fb,77,00,00,07,00,90,b5,e8,01,00,00,00,00,78,01,07,\
  00,98,74,31,05,00,00,07,00,13,9e,fc,77,08,06,07,00,8f,9d,fc,77,01,00,00,00,\
  e4,fb,d2,06,54,53,f1,01,41,9d,fc,77,00,00,07,00,13,9e,fc,77,08,06,07,00,8f,\
  9d,fc,77,01,00,00,00,00,00,07,00,a0,b5,e8,01,ff,ff,ff,ff,40,53,f1,01,cc,fa,\
  d2,06,00,36,f1,01,03,00,00,00,88,03,07,00,88,03,07,00,08,00,00,00,04,00,00,\
  00,88,03,07,00,02,00,00,00,02,00,00,00,00,02,07,00,00,02,07,00,01,01,07,00,\
  02,00,00,00,90,b5,e8,01,01,fb,d2,06,db,80,fb,77,70,fb,d2,06,00,00,00,00,c4,\
  fa,d2,06,00,00,00,00,6e,b5,f8,77,00,00,00,00,48,fb,d2,06,70,7c,db,77,00,00,\
  00,00,00,56,4b,6a,02,00,00,80,18,00,00,00,00,00,00,00,40,53,f1,01,00,00,00,\
  00,01,00,00,00,00,00,07,00,c8,95,31,05,00,00,00,00,a0,fb,d2,06,41,9d,fc,77,\
  00,00,07,00,13,9e,fc,77,08,06,07,00,8f,9d,fc,77,a4,7e,e5,01,00,00,00,00,08,\
  36,f1,01,00,00,00,00,00,00,00,00,00,00,00,00,80,fb,d2,06,0f,94,db,77,ec,08,\
  00,00,54,fb,d2,06,03,00,00,00,d0,95,31,05,98,74,31,05,78,01,07,00,98,74,31,\
  05,78,01,07,00,90,74,31,05,0c,00,0e,00,ac,58,4b,6a,00,00,00,00,c1,2b,4d,6a,\
  00,00,00,00,44,fb,d2,06,00,00,07,00,a6,00,00,00,01,19,db,77

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000006]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
```

  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0e,00,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\
  48,a1,92,ee,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,05,00,00,00,00,00,00,80,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,32,00,36,00,32,00,41,00,34,00,45,00,31,00,32,00,2d,00,44,00,\
  33,00,39,00,42,00,2d,00,34,00,36,00,42,00,31,00,2d,00,41,00,45,00,41,00,37,\
  00,2d,00,46,00,31,00,42,00,30,00,45,00,45,00,38,00,41,00,41,00,35,00,31,00,\
  44,00,7d,00,5d,00,20,00,53,00,45,00,51,00,50,00,41,00,43,00,4b,00,45,00,54,\
  00,20,00,30,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000007]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,02,02,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,08,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\
  48,a1,92,ef,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,02,00,00,00,00,00,00,80,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,32,00,36,00,32,00,41,00,34,00,45,00,31,00,32,00,2d,00,44,00,\
  33,00,39,00,42,00,2d,00,34,00,36,00,42,00,31,00,2d,00,41,00,45,00,41,00,37,\
  00,2d,00,46,00,31,00,42,00,30,00,45,00,45,00,38,00,41,00,41,00,35,00,31,00,\
  44,00,7d,00,5d,00,20,00,44,00,41,00,54,00,41,00,47,00,52,00,41,00,4d,00,20,\
  00,30,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000008]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0e,00,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\
  48,a1,92,f0,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,05,00,00,00,ff,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,32,00,31,00,43,00,44,00,35,00,35,00,37,00,34,00,2d,00,31,00,\
  42,00,41,00,35,00,2d,00,34,00,39,00,36,00,38,00,2d,00,41,00,31,00,35,00,36,\
  00,2d,00,45,00,42,00,30,00,46,00,38,00,35,00,36,00,43,00,43,00,38,00,34,00,\
  44,00,7d,00,5d,00,20,00,53,00,45,00,51,00,50,00,41,00,43,00,4b,00,45,00,54,\
  00,20,00,31,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000009]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,02,02,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\

```
  48,a1,92,f1,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,02,00,00,00,ff,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,32,00,31,00,43,00,44,00,35,00,35,00,37,00,34,00,2d,00,31,00,\
  42,00,41,00,35,00,2d,00,34,00,39,00,36,00,38,00,2d,00,41,00,31,00,35,00,36,\
  00,2d,00,45,00,42,00,30,00,46,00,38,00,35,00,36,00,43,00,43,00,38,00,34,00,\
  44,00,7d,00,5d,00,20,00,44,00,41,00,54,00,41,00,47,00,52,00,41,00,4d,00,20,\
  00,31,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000010]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,0e,00,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\
  48,a1,92,f2,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,05,00,00,00,fe,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,30,00,44,00,39,00,32,00,42,00,34,00,31,00,42,00,2d,00,37,00,\
  32,00,39,00,31,00,2d,00,34,00,34,00,33,00,42,00,2d,00,42,00,36,00,46,00,31,\
  00,2d,00,39,00,34,00,46,00,44,00,36,00,39,00,46,00,42,00,32,00,45,00,34,00,\
  41,00,7d,00,5d,00,20,00,53,00,45,00,51,00,50,00,41,00,43,00,4b,00,45,00,54,\
  00,20,00,32,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\Protocol_C
atalog9\Catalog_Entries\000000000011]
"PackedCatalogItem"=hex:25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,\
  6d,33,32,5c,6d,73,61,66,64,2e,64,6c,6c,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,09,02,02,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,30,18,5f,8d,73,c2,cf,11,95,c8,00,80,5f,\
  48,a1,92,f3,03,00,00,01,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,02,00,00,00,11,00,00,00,14,00,00,\
  00,14,00,00,00,02,00,00,00,fe,ff,ff,ff,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,fa,00,00,00,00,00,00,4d,00,53,00,41,00,46,00,44,00,20,00,4e,00,65,00,74,\
  00,42,00,49,00,4f,00,53,00,20,00,5b,00,5c,00,44,00,65,00,76,00,69,00,63,00,\
  65,00,5c,00,4e,00,65,00,74,00,42,00,54,00,5f,00,54,00,63,00,70,00,69,00,70,\
  00,5f,00,7b,00,30,00,44,00,39,00,32,00,42,00,34,00,31,00,42,00,2d,00,37,00,\
  32,00,39,00,31,00,2d,00,34,00,34,00,33,00,42,00,2d,00,42,00,36,00,46,00,31,\
  00,2d,00,39,00,34,00,46,00,44,00,36,00,39,00,46,00,42,00,32,00,45,00,34,00,\
  41,00,7d,00,5d,00,20,00,44,00,41,00,54,00,41,00,47,00,52,00,41,00,4d,00,20,\
  00,32,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,\
  00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\SubjectPackages]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\SubjectPackages\MS
Subjects 1]
"$DLL"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,4d,73,53,69,70,31,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\SubjectPackages\MS
Subjects 2]
"$DLL"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,4d,73,53,69,70,32,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\SubjectPackages\MS
Subjects 3]
"$DLL"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,4d,73,53,69,70,33,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\TrustProviders]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinTrust\TrustProviders\Softwa
re Publisher]
```

```
"$DLL"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,32,\
  5c,53,6f,66,74,50,75,62,2e,64,6c,6c,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wmi]
"Description"="Provides systems management information to and from drivers."
"DisplayName"="Windows Management Instrumentation Driver Extensions"
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,73,79,73,74,65,6d,33,\
  32,5c,53,65,72,76,69,63,65,73,2e,65,78,65,00
"ObjectName"="LocalSystem"
"Start"=dword:00000003
"Type"=dword:00000020

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wmi\Security]
"Security"=hex:01,00,14,80,b4,00,00,00,c0,00,00,00,14,00,00,00,34,00,00,00,02,\
  00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,20,02,00,00,02,00,80,00,05,00,00,00,00,03,18,00,8d,00,02,00,01,01,00,\
  00,00,00,00,01,00,00,00,00,00,00,00,00,00,03,18,00,ff,01,0f,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,20,02,00,00,00,03,18,00,8f,00,02,00,01,02,00,00,00,\
  00,00,05,20,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01,01,00,00,00,00,\
  00,05,04,00,00,00,23,02,00,00,00,03,18,00,9d,00,00,00,01,02,00,00,00,00,00,\
  05,20,00,00,00,21,02,00,00,01,01,00,00,00,00,00,05,12,00,00,00,01,01,00,00,\
  00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Wmi\Enum]
"0"="Root\\LEGACY_WMI\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\{262A4E12-D39B-46B1-AEA7-
F1B0EE8AA51D}]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\{262A4E12-D39B-46B1-AEA7-
F1B0EE8AA51D}\Parameters]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\{262A4E12-D39B-46B1-AEA7-
F1B0EE8AA51D}\Parameters\Tcpip]
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,33,30,2e,31,36,38,2e,32,30,36,2e,39,39,00,00
"SubnetMask"=hex(7):32,35,35,2e,32,35,35,2e,30,2e,30,00,00
"DefaultGateway"=hex(7):00
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:398ed4c1
"T1"=dword:398edbc9
"T2"=dword:398ee10f
"LeaseTerminatesTime"=dword:398ee2d1
```

TPCC Application Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TPCC]
"Path"="c:\\inetpub\\wwwroot\\"
"NumberOfDeliveryThreads"=dword:0000000a
"MaxConnections"=dword:00002710
"MaxPendingDeliveries"=dword:000003e8
"DB_Protocol"="ODBC"
"TxnMonitor"="COM"
"DbServer"="cheapo_ip"
"DbName"="tpcc"
"DbUser"="sa"
"DbPassword"=""
"COM_SinglePool"="YES"
```

Server Bus Performance Driver Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb]
"Type"=dword:00000001
"Start"=dword:00000000
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,70,71,\
  63,69,73,73,62,2e,73,79,73,00
"DisplayName"="Compaq CISS Controllers Device Driver"
"Group"="port"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Parameters]
"CompletionMode"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,00,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,00,00,00,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissb\Enum]
"0"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&267a616a&0&38"
"Count"=dword:00000003
"NextInstance"=dword:00000003
"1"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&13c0b0c5&0&40"
"2"="PCI\\VEN_0E11&DEV_B060&SUBSYS_40700E11&REV_02\\3&1070020&0&38"
```

Server Disk Device Performance Driver Registry Parameters

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd]
"Type"=dword:00000001
"Start"=dword:00000000
```

```
"ErrorControl"=dword:00000001
"Tag"=dword:00000102
"ImagePath"=hex(2):53,79,73,74,65,6d,33,32,5c,44,52,49,56,45,52,53,5c,63,70,71,\
  63,69,73,73,64,2e,73,79,73,00
"DisplayName"="Compaq CISS Controllers Disk Driver"
"Group"="Primary Disk"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,00,00,00,14,00,00,00,30,00,00,00,02,\
  00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,02,00,70,00,04,00,00,00,00,00,18,00,fd,01,02,00,01,01,00,00,00,00,00,\
  05,12,00,00,00,50,00,5f,00,00,00,1c,00,ff,01,0f,00,01,02,00,00,00,00,00,05,\
  20,00,00,00,20,02,00,00,31,00,33,00,00,00,18,00,8d,01,02,00,01,01,00,00,00,\
  00,00,05,0b,00,00,00,20,02,00,00,00,1c,00,fd,01,02,00,01,02,00,00,00,00,00,\
  00,05,20,00,00,00,23,02,00,00,31,00,33,00,01,01,00,00,00,00,00,05,12,00,00,\
  00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\cpqcissd\Enum]
"0"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0000004000000000"
"Count"=dword:00000009
"NextInstance"=dword:00000009
"1"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0100004000000000"
"2"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&3b708dbf&0&0200004000000000"
"3"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0000004000000000"
"4"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0100004000000000"
"5"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&2b81de8b&0&0200004000000000"
"6"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0000004000000000"
"7"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0100004000000000"
"8"="CPQCISS\\Disk&VEN_COMPAQ&PROD_LOGICAL_VOLUME\\4&161bf83a&0&0200004000000000"
```

Client System Configuration

```
Date . . . . . .  08/14/2000
Time . . . . . .  08:38:04

Product  . . . . . .  ProLiant ML350

Machine ID
  From System Board  . . . . . .  653

Processor  . . . . . .  Pentium III(R) at 733 MHz
  Slot . . . . . .  1
  Secondary Cache  . . . . . .  256K
  CPU ID . . . . . .  0681


Numeric Coprocessor  . . . . . .  Integrated 387-Compatible

Expansion Bus  . . . . . .  ISA, PCI

System Identification Number . .  0

CPU Mode . . . . . .  Real Mode

Current System Speed . . . . . .  High

System ROM
    Revision . . . . . .  06/17/2000
      Family . . . . . .  D2
    Flashable  . . . . . .  Yes
    Supports F10 partition . . .  Yes
    Socketed . . . . . .  Yes

Video Controller ROM
    Revision . . . . . .  3.96

Option ROMs
  Address Range  . . . . . .  C0000 - C7FFF
    Data Dump  . . . . . .  (1999/03/03 15:43)

  Address Range  . . . . . .  C8000 - CFFFF
    Data Dump  . . . . . .  ((06/17/2000) Compaq Server Feature Board BIOS
Vers...)

  Address Range  . . . . . .  D0000 - D0FFF
    Data Dump  . . . . . .  (  Copyright (C) 1997,1998  Intel Corporation  )

  Address Range  . . . . . .  E0000 - EFFFF

Bootblock ROM  . . . . . .  12/16/1999
Memory Boards Identified:
  System Board
    DIMM Slot  1 (SDRAM) . . . .   128 Megabytes
    DIMM Slot  2 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  3 . . . . . .     0 Megabytes
    DIMM Slot  4 . . . . . .     0 Megabytes
Total Compaq Memory  . . . . . .  384 Megabytes

Keyboard . . . . . .  Enhanced

LPT Ports  . . . . . .  LPT1 (Address 378)

COM Ports  . . . . . .  COM1 (Address 3F8)
                        COM2 (Address 2F8)
Compaq NC3163 Fast Ethernet NIC
  Device Type  . . . . . .  Ethernet Controller
  PCI Bus Number . . . . . .  1
  Device Number  . . . . . .  5
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  1229h
  Subsystem Vendor ID  . . . . .  0E11h
  Subsystem ID . . . . . .  B134h
  Revision ID  . . . . . .  08h
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . .  11
  IRQ Pin  . . . . . .  INTA#
  Memory Address Base  . . . . .  B0500000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . .  2000h
  IO Address Length  . . . . . .  40h
  Memory Address Base  . . . . .  B0400000h
  Memory Address Length  . . . .  100000h


Compaq NC3123 Fast Ethernet NIC
  Device Type  . . . . . .  Ethernet Controller
  PCI Bus Number . . . . . .  5
  Device Number  . . . . . .  1
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  2
```

```
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  1229h
  Subsystem Vendor ID  . . . . .  0E11h
  Subsystem ID . . . . . .  B144h
  Revision ID  . . . . . .  08h
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . .  5
  IRQ Pin  . . . . . .  INTA#
  Memory Address Base  . . . . .  D0100000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . .  B000h
  IO Address Length  . . . . . .  40h
  Memory Address Base  . . . . .  D0000000h
  Memory Address Length  . . . .  100000h



Diskette Drive A . . . . . .  1.44 Megabyte (3.5 inch)

Graphics Mode  . . . . . .  03 (80-Column Text)


Primary Monitor attached to  . .  ATI RAGE IIC PCI Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . .    640 Kbytes
  Amount Free  . . . . . .    597 Kbytes (612144 Bytes)

Extended Memory
  System Total . . . . . .  392192 Kbytes

Expanded Memory
  LIM Driver Support . . . . . .  LIM driver not loaded

Operating System . . . . . .  MS-DOS version 7.00 (from diskette)

Environment variables
  PATH=
  PROMPT=$P$G
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

System serial number . . . . . .  0

Memory Allocation (including INSPECT)
  PSP   SIZE    NAME          TRAPPED INTERRUPTS
  ----  ------  ------------  ---------------------------------------
  08C8  007200  COMMAND.COM   2Fh  2Eh  24h  23h  22h
  0A93  218144  INSPECT.EXE   3Fh  00h

System Configuration Memory
  00 - 0F :    15 5A 38 00    08 5A 01 14    08 00 26 82    50 80 00 00
  10 - 1F :    40 00 00 00    03 80 02 00    FC 00 00 00    00 F0 00 05
  20 - 2F :    00 00 00 00    7E 2B 00 40    00 9E 02 60    40 00 04 DF
  30 - 3F :    00 FC 20 80    00 00 XX XX    XX XX XX XX    XX XX XX XX

BIOS Data Area
  40:0000 :    F8 03 F8 02    00 00 00 00    78 03 00 00    00 00 04 02
  40:0010 :    27 44 00 80    02 00 00 20    00 00 1E 00    1E 00 00 00
  40:0020 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:0030 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 01 01
  40:0040 :    25 00 04 00    00 2B 01 0F    02 03 50 00    00 10 00 00
  40:0050 :    00 18 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:0060 :    0E 0D 00 D4    03 29 30 C2    11 45 77 00    B1 A2 08 00
  40:0070 :    00 00 00 12    00 01 00 00    14 14 14 14    01 01 01 01
  40:0080 :    1E 00 3E 00    18 10 00 60    F9 11 0B 01    00 00 00 05
  40:0090 :    17 00 00 00    2B 00 10 12    00 00 00 00    00 00 00 00
  40:00A0 :    00 00 00 00    00 00 00 00    7C 14 00 C0    00 00 00 00
  40:00B0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00C0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00D0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00E0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00F0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00


Interrupt Vector Table (including INSPECT)
  00 - 03 :    0AA3:0555      0070:0465      07FF:0016      0070:0465
  04 - 07 :    0070:0465      F000:FF54      F000:E7C8      F000:9BD0
  08 - 0B :    07FF:001F      07FF:0028      F000:9BD0      F000:9BD0
  0C - 0F :    F000:9BD0      F000:9BD0      07FF:009A      0070:0465
  10 - 13 :    C000:13FE      F000:F84D      F000:F841      0070:03EE
  14 - 17 :    F000:E739      0245:0240      0070:042D      F000:EFD2
  18 - 1B :    F000:A301      08C0:002F      F000:FE6E      0070:045F
  1C - 1F :    F000:FF53      F000:0000      0000:0522      C000:2143
  20 - 23 :    00C9:0FA8      00C9:0FB2      08C8:0314      08C8:016D
  24 - 27 :    08C8:0178      00C9:0FBC      00C9:0FC6      00C9:0FD0
  28 - 2B :    00C9:106F      0070:0466      00C9:106F      00C9:106F
  2C - 2F :    00C9:106F      00C9:106F      08C8:0162      08C9:01CC
  30 - 33 :    C90F:E4EA      F000:9B00      00C9:106F      00C9:106F
  34 - 37 :    00C9:106F      00C9:106F      00C9:106F      00C9:106F
  38 - 3B :    00C9:106F      00C9:106F      00C9:106F      00C9:106F
  3C - 3F :    00C9:106F      00C9:106F      00C9:106F      1B5F:04F3
  40 - 43 :    F000:9175      0000:0000      F000:F065      C000:2556
  44 - 47 :    F000:9BD0      F000:9BD0      0000:0000      F000:9BD0
  48 - 4B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  4C - 4F :    F000:9BD0      F000:9BD0      F000:9BD0      0070:04FC
  50 - 53 :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  54 - 57 :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  58 - 5B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  5C - 5F :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  60 - 63 :    0000:0000      0000:0000      0000:0000      0000:0000
  64 - 67 :    0000:0000      0000:0000      0000:0000      0000:0000
  68 - 6B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  6C - 6F :    F000:9BD0      C000:13FE      F000:9BD0      F000:9BD0
  70 - 73 :    07FF:0035      F000:9C1F      F000:9BD0      F000:9BD0
  74 - 77 :    07FF:00E2      F000:9C28      07FF:00FA      07FF:0112
  78 - 7B :    0000:0000      0000:0000      0000:0000      0000:0000
  7C - 7F :    0000:0000      0000:0000      0000:0000      0000:0000
  80 - 83 :    0000:0000      0000:0000      0000:0000      0000:0000
  84 - 87 :    0000:0000      0000:0000      0000:0000      0000:0000
  88 - 8B :    0000:0000      0000:0000      0000:0000      0000:0000
  8C - 8F :    0000:0000      0000:0000      0000:0000      0000:0000
  90 - 93 :    0000:0000      0000:0000      0000:0000      0000:0000
  94 - 97 :    0000:0000      0000:0000      0000:0000      0000:0000
  98 - 9B :    0000:0000      0000:0000      0000:0000      0000:0000
  9C - 9F :    0000:0000      0000:0000      0000:0000      0000:0000
  A0 - A3 :    0000:0000      0000:0000      0000:0000      0000:0000
  A4 - A7 :    0000:0000      0000:0000      0000:0000      0000:0000
  A8 - AB :    0000:0000      0000:0000      0000:0000      0000:0000
  AC - AF :    0000:0000      0000:0000      0000:0000      0000:0000
  B0 - B3 :    0000:0000      0000:0000      0000:0000      0000:0000
  B4 - B7 :    0000:0000      0000:0000      0000:0000      0000:0000
  B8 - BB :    0000:0000      0000:0000      0000:0000      0000:0000
```

```
  BC - BF :    0000:0000      0000:0000      0000:0000      0000:0000
  C0 - C3 :    0000:0000      0000:0000      0000:0000      0000:0000
  C4 - C7 :    0000:0000      0000:0000      0000:0000      0000:0000
  C8 - CB :    0000:0000      0000:0000      0000:0000      0000:0000
  CC - CF :    0000:0000      0000:0000      0000:0000      0000:0000
  D0 - D3 :    0000:0000      0000:0000      0000:0000      0000:0000
  D4 - D7 :    0000:0000      0000:0000      0000:0000      0000:0000
  D8 - DB :    0000:0000      0000:0000      0000:0000      0000:0000
  DC - DF :    0000:0000      0000:0000      0000:0000      0000:0000
  E0 - E3 :    0000:0000      0000:0000      0000:0000      0000:0000
  E4 - E7 :    0000:0000      0000:0000      0000:0000      0000:0000
  E8 - EB :    0000:0000      0000:0000      0000:0000      0000:0000
  EC - EF :    0000:0000      0000:0000      0000:0000      0000:0000
  F0 - F3 :    0000:0000      0000:0000      0000:0000      0000:0000
  F4 - F7 :    0000:0000      0000:0000      0000:0000      0000:0000
  F8 - FB :    0000:0000      0000:0000      0000:0000      0000:0000
  FC - FF :    0000:0000      0000:0000      0000:0000      0000:0000

PCI Devices Information
  Signature  . . . . . .  PCI
  Config Mechanism #1  . . . . .  Supported
  Config Mechanism #2  . . . . .  Not Supported
  Spec Cycle for Config #1 . . .  Supported
  Spec Cycle for Config #2 . . .  Not Supported
  BIOS Interface Version . . . .  2.10
  Last PCI Bus Number  . . . . .  5
  Number of PCI Devices  . . . .  6

  PCI Bus Number . . . . . .  0
  Device Number  . . . . . .  15
  Function Number  . . . . . .  01h
  Slot Number  . . . . . .  0
  Vendor ID  . . . . . .  1166h
  Device ID  . . . . . .  0211h
  Revision ID  . . . . . .  00h
  Device Type  . . . . . .  IDE Controller
  Programming Interface  . . . .  8Ah
  Expansion ROM Base Address . .  0h
  IRQ Line . . . . . .  0
  IRQ Pin  . . . . . .  Not Used
  IO Address Base  . . . . . .  3000h
  IO Address Length  . . . . . .  10h

  PCI Bus Number . . . . . .  1
  Device Number  . . . . . .  4
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  000Bh
  Revision ID  . . . . . .  07h
  Device Type  . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFF0000h
  IRQ Line . . . . . .  15
  IRQ Pin  . . . . . .  INTA#
  IO Address Base  . . . . . .  1000h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  B0200000h
  Memory Address Length  . . . .  400h
  Memory Address Base  . . . . .  B0000000h
  Memory Address Length  . . . .  2000h

  PCI Bus Number . . . . . .  1
  Device Number  . . . . . .  4
  Function Number  . . . . . .  01h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  000Bh
  Revision ID  . . . . . .  07h
  Device Type  . . . . . .  SCSI Bus Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFF0000h
  IRQ Line . . . . . .  15
  IRQ Pin  . . . . . .  INTA#
  IO Address Base  . . . . . .  1400h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  B0300000h
  Memory Address Length  . . . .  400h
  Memory Address Base  . . . . .  B0100000h
  Memory Address Length  . . . .  2000h

  PCI Bus Number . . . . . .  1
  Device Number  . . . . . .  5
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  1229h
  Revision ID  . . . . . .  08h
  Device Type  . . . . . .  Ethernet Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . .  11
  IRQ Pin  . . . . . .  INTA#
  Memory Address Base  . . . . .  B0500000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . .  2000h
  IO Address Length  . . . . . .  40h
  Memory Address Base  . . . . .  B0400000h
  Memory Address Length  . . . .  100000h

  PCI Bus Number . . . . . .  1
  Device Number  . . . . . .  6
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  1002h
  Device ID  . . . . . .  4756h
  Revision ID  . . . . . .  7Ah
  Device Type  . . . . . .  VGA Compatible Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFFE0000h
  IRQ Line . . . . . .  0
  IRQ Pin  . . . . . .  Not Used
  Memory Address Base  . . . . .  B1000000h
  Memory Address Length  . . . .  1000000h
  IO Address Base  . . . . . .  1800h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  B0600000h
  Memory Address Length  . . . .  1000h

  PCI Bus Number . . . . . .  5
  Device Number  . . . . . .  1
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  2
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  1229h
  Revision ID  . . . . . .  08h
```

```
  Device Type  . . . . . .  Ethernet Controller
  Programming Interface  . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . .  5
  IRQ Pin  . . . . . .  INTA#
  Memory Address Base  . . . . .  D0100000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . .  B000h
  IO Address Length  . . . . . .  40h
  Memory Address Base  . . . . .  D0000000h
  Memory Address Length  . . . .  100000h
```


SUT System Configuration

```
Date . . . . . .  08/14/2000
Time . . . . . .  08:47:12

Product  . . . . . .  ProLiant ML570

Machine ID
  From System Board  . . . . . .  CPQ0715

Processor  . . . . . .  Pentium III(R) Xeon at 700 MHz
  Slot . . . . . .  3
  Secondary Cache  . . . . . .  2048K
  CPU ID . . . . . .  06A0

Processor  . . . . . .  Pentium III(R) Xeon at 700 MHz
  Slot . . . . . .  2
  Secondary Cache  . . . . . .  1024K
  CPU ID . . . . . .  06A1

Processor  . . . . . .  Pentium III(R) Xeon at 700 MHz
  Slot . . . . . .  1
  Secondary Cache  . . . . . .  1024K
  CPU ID . . . . . .  06A1

Processor(s) Mapped Out  . . . .  None

Numeric Coprocessor  . . . . . .  Integrated 387-Compatible

Expansion Bus  . . . . . .  ISA, PCI

System Identification Number . .  0013DYV10026

CPU Mode . . . . . .  Real Mode

System ROM
    Revision . . . . . .  12/27/1999
      Family . . . . . .  P20
    Flashable  . . . . . .  Yes
    Supports F10 partition . . .  Yes

Video Controller ROM
    Revision . . . . . .  3.96

Option ROMs
  Address Range  . . . . . .  C0000 - C7FFF
    Data Dump  . . . . . .  (1999/03/24 23:56)

  Address Range  . . . . . .  C8000 - CBFFF
    Data Dump  . . . . . .  (04/22/98 ROC     Smart Array Option ROM/BIOS
(C)Co...)

  Address Range  . . . . . .  CC000 - CFFFF
    Data Dump  . . . . . .  (07/07/00 Maxwell Smart Array Option ROM/BIOS
(C)Co...)

  Address Range  . . . . . .  D0000 - D17FF
    Data Dump  . . . . . .  (  Copyright (C) 1997,1998,1999  Intel Corporation
...)

  Address Range  . . . . . .  E8000 - EDFFF
    Data Dump  . . . . . .  ( CPQSCSI d)

Bootblock ROM  . . . . . .  02/28/2000
Memory Boards Identified:
  System Board
    DIMM Slot  1 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  2 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  3 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  4 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  5 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  6 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  7 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  8 (SDRAM) . . . .   256 Megabytes
    DIMM Slot  9 (SDRAM) . . . .   128 Megabytes
    DIMM Slot  10 (SDRAM)  . . .   128 Megabytes
    DIMM Slot  11 (SDRAM)  . . .   128 Megabytes
    DIMM Slot  12 (SDRAM)  . . .   128 Megabytes
    DIMM Slot  13  . . . . . .     0 Megabytes
    DIMM Slot  14  . . . . . .     0 Megabytes
    DIMM Slot  15  . . . . . .     0 Megabytes
    DIMM Slot  16  . . . . . .     0 Megabytes
Total Compaq Memory  . . . . . .  2560 Megabytes

Keyboard . . . . . .  Enhanced

LPT Ports  . . . . . .  LPT1 (Address 3BC)

COM Ports  . . . . . .  COM1 (Address 3F8)
Compaq NC3123 Fast Ethernet NIC
  Device Type  . . . . . .  Ethernet Controller
  PCI Bus Number . . . . . .  2
  Device Number  . . . . . .  6
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  1
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  1229h
  Subsystem Vendor ID  . . . . .  0E11h
  Subsystem ID . . . . . .  B144h
  Revision ID  . . . . . .  08h
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF00000h
  IRQ Line . . . . . .  10
  IRQ Pin  . . . . . .  INTA#
  Memory Address Base  . . . . .  F2DE0000h
  Memory Address Length  . . . .  1000h
  IO Address Base  . . . . . .  3000h
```

```
  IO Address Length  . . . . . .  40h
  Memory Address Base  . . . . .  F2C00000h
  Memory Address Length  . . . .  100000h



Diskette Drive A . . . . . .  1.44 Megabyte (3.5 inch)

Drive Controller 1, Compaq Integrated Smart Array Controller
  IDA Firmware Revision  . . . .  1.42
  Array Accelerator Memory . . .  8188 Kbytes
    Reserved for reads . . . . .  8188 Kbytes
  Accelerator Status . . . . . .  Not Configured
    Battery count  . . . . . .  0
    Batteries charged  . . . . .  0
    Batteries failed . . . . . .  0
  Internal ProLiant  . . . . . .  Bus 1, Rev. JB21
  Internal ProLiant  . . . . . .  Bus 2, Rev. JB21

  Logical Drive  1 . . . . . .  9091 Megabyte
  Fault Tolerance  . . . . . .  None
  OS Format  . . . . . .  Multi-Sector Distribution
  Drive geometry (Cyl, Hds, Sec)  2176, 255, 32
  Array Accelerator  . . . . . .  Disabled

    Hard Drive  1
    SCSI Bus . . . . . .  1
    SCSI ID  . . . . . .  0
    Serial Number  . . . . . .  LS811438000010211C0Q
    Firmware Revision 1  . . . .  3B07
    Model Number . . . . . .  COMPAQ  BD009122BA
    Initialized for Monitoring .  Yes
      Reference time . . . . . .  8472
      Sectors read . . . . . .  2637400
      Hard read errors . . . . .  0
      Read errors retry  . . . .  0
      ECC read errors  . . . . .  0
      Sectors written  . . . . .  2055958
      Hard write errors  . . . .  0
      Write errors retry . . . .  0
      Seek count . . . . . .  37840
      Seek errors  . . . . . .  0
      Spin cycles  . . . . . .  0
      Spin up time . . . . . .  0
      Seek time track  . . . . .  0%
      Seek time third  . . . . .  0%
      Seek time full . . . . . .  0%
      Reallocated sectors  . . .  442
      Recovers read failed . . .  0
      Bus faults . . . . . .  0

  Logical Drive  2 . . . . . .  54621 Megabyte
  Fault Tolerance  . . . . . .  Mirroring
  OS Format  . . . . . .  Multi-Sector Distribution
  Drive geometry (Cyl, Hds, Sec)  13074, 255, 32
  Array Accelerator  . . . . . .  Disabled

    Hard Drive  1
    SCSI Bus . . . . . .  1
    SCSI ID  . . . . . .  3
    Serial Number  . . . . . .  B2132446
    Firmware Revision 1  . . . .  B016
    Model Number . . . . . .  COMPAQ  BD018122C9
    Initialized for Monitoring .  Yes
      Reference time . . . . . .  89150
      Sectors read . . . . . .  *642779168
      Hard read errors . . . . .  0
      Read errors retry  . . . .  0
      ECC read errors  . . . . .  0
      Sectors written  . . . . .  1181474008
      Hard write errors  . . . .  0
      Write errors retry . . . .  0
      Seek count . . . . . .  374477
      Seek errors  . . . . . .  0
      Spin cycles  . . . . . .  0
      Spin up time . . . . . .  0
      Seek time track  . . . . .  0%
      Seek time third  . . . . .  0%
      Seek time full . . . . . .  0%
      Reallocated sectors  . . .  6
      Recovers read failed . . .  0
      Bus faults . . . . . .  0

    Hard Drive  2
    SCSI Bus . . . . . .  1
    SCSI ID  . . . . . .  4
    Serial Number  . . . . . .  B2134058
    Firmware Revision 1  . . . .  B016
    Model Number . . . . . .  COMPAQ  BD018122C9
    Initialized for Monitoring .  Yes
      Reference time . . . . . .  78089
      Sectors read . . . . . .  *2612564036
      Hard read errors . . . . .  0
      Read errors retry  . . . .  0
      ECC read errors  . . . . .  0
      Sectors written  . . . . .  1169918519
      Hard write errors  . . . .  0
      Write errors retry . . . .  0
      Seek count . . . . . .  329401
      Seek errors  . . . . . .  0
      Spin cycles  . . . . . .  0
      Spin up time . . . . . .  0
      Seek time track  . . . . .  0%
      Seek time third  . . . . .  0%
      Seek time full . . . . . .  0%
      Reallocated sectors  . . .  66
      Recovers read failed . . .  0
      Bus faults . . . . . .  0

    Hard Drive  3
    SCSI Bus . . . . . .  1
    SCSI ID  . . . . . .  5
    Serial Number  . . . . . .  B2133805
    Firmware Revision 1  . . . .  B016
    Model Number . . . . . .  COMPAQ  BD018122C9
    Initialized for Monitoring .  Yes
      Reference time . . . . . .  89151
      Sectors read . . . . . .  *3074536954
      Hard read errors . . . . .  0
      Read errors retry  . . . .  0
      ECC read errors  . . . . .  0
      Sectors written  . . . . .  780217809
      Hard write errors  . . . .  0
      Write errors retry . . . .  0
      Seek count . . . . . .  363847
      Seek errors  . . . . . .  0
```

```
  Spin cycles  . . . . . .  0
  Spin up time . . . . . .  0
  Seek time track  . . . .  0%
  Seek time third  . . . .  0%
  Seek time full . . . . .  0%
  Reallocated sectors  . .  69
  Recovers read failed . .  0
  Bus faults . . . . . .  0

Hard Drive  4
SCSI Bus . . . . . .  2
SCSI ID  . . . . . .  3
Serial Number  . . . . . .  B2134244
Firmware Revision 1  . . . .  B016
Model Number . . . . . .  COMPAQ  BD018122C9
Initialized for Monitoring .  Yes
  Reference time . . . . . .  63155
  Sectors read . . . . . .  *174234886
  Hard read errors . . . . .  0
  Read errors retry  . . . .  0
  ECC read errors  . . . . .  0
  Sectors written  . . . . .  616160864
  Hard write errors  . . . .  0
  Write errors retry . . . .  0
  Seek count . . . . . .  278631
  Seek errors  . . . . . .  0
  Spin cycles  . . . . . .  0
  Spin up time . . . . . .  0
  Seek time track  . . . . .  40%
  Seek time third  . . . . .  73%
  Seek time full . . . . . .  73%
  Reallocated sectors  . . .  18
  Recovers read failed . . .  0
  Bus faults . . . . . .  0

Hard Drive  5
SCSI Bus . . . . . .  2
SCSI ID  . . . . . .  4
Serial Number  . . . . . .  B2126326
Firmware Revision 1  . . . .  B016
Model Number . . . . . .  COMPAQ  BD018122C9
Initialized for Monitoring .  Yes
  Reference time . . . . . .  85040
  Sectors read . . . . . .  *1119960730
  Hard read errors . . . . .  0
  Read errors retry  . . . .  0
  ECC read errors  . . . . .  0
  Sectors written  . . . . .  692470080
  Hard write errors  . . . .  0
  Write errors retry . . . .  0
  Seek count . . . . . .  343389
  Seek errors  . . . . . .  0
  Spin cycles  . . . . . .  0
  Spin up time . . . . . .  0
  Seek time track  . . . . .  0%
  Seek time third  . . . . .  0%
  Seek time full . . . . . .  0%
  Reallocated sectors  . . .  97
  Recovers read failed . . .  0
  Bus faults . . . . . .  0

Hard Drive  6
SCSI Bus . . . . . .  2
  SCSI ID  . . . . . .  5
  Serial Number  . . . . . .  B2134471
  Firmware Revision 1  . . . .  B016
  Model Number . . . . . .  COMPAQ  BD018122C9
  Initialized for Monitoring .  Yes
    Reference time . . . . . .  73867
    Sectors read . . . . . .  *1121163839
    Hard read errors . . . . .  0
    Read errors retry  . . . .  0
    ECC read errors  . . . . .  0
    Sectors written  . . . . .  696718272
    Hard write errors  . . . .  0
    Write errors retry . . . .  0
    Seek count . . . . . .  311040
    Seek errors  . . . . . .  0
    Spin cycles  . . . . . .  0
    Spin up time . . . . . .  0
    Seek time track  . . . . .  36%
    Seek time third  . . . . .  72%
    Seek time full . . . . . .  73%
    Reallocated sectors  . . .  117
    Recovers read failed . . .  0
    Bus faults . . . . . .  0

Graphics Mode  . . . . . .  03 (80-Column Text)


Primary Monitor attached to  . .  ATI RAGE IIC PCI Graphics Controller
with Video Graphics Color Monitor

Base Memory
  System Total . . . . . .    636 Kbytes
  Amount Free  . . . . . .    594 Kbytes (608960 Bytes)

Extended Memory
  System Total . . . . . .  2620416 Kbytes

Expanded Memory
  LIM Driver Support . . . . . .  LIM driver not loaded

Operating System . . . . . .  MS-DOS version 7.00 (from diskette)

Environment variables
  PATH=
  PROMPT=$P$G
  COMSPEC=A:\COMMAND.COM
  CMDLINE=inspect /u
End of environment

Revisions Table

Previous Revisions

Current Revisions

System serial number . . . . . .  0013DYV10026

Memory Allocation (including INSPECT)
  PSP   SIZE    NAME          TRAPPED INTERRUPTS
  ----  ------  ------------  ----------------------------------
  088F  007200  COMMAND.COM   ECh  2Fh  2Eh  24h  23h  22h
  0A5A  218144  INSPECT.EXE   F9h  F4h  F3h  F2h  EFh  EDh  3Fh  00h
```

```
System Configuration Memory
  00 - 0F :    18 00 47 00    08 00 01 14    08 00 26 82    50 80 00 00
  10 - 1F :    40 00 00 00    03 80 02 00    3C 00 00 00    00 00 00 02
  20 - 2F :    00 00 00 00    7F 20 20 40    00 9A 00 00    00 10 02 AC
  30 - 3F :    00 3C 20 80    00 00 XX XX    XX XX XX XX    XX XX XX XX

BIOS Data Area
  40:0000 :    F8 03 00 00    00 00 00 00    BC 03 00 00    00 00 00 9F
  40:0010 :    27 42 00 7C    02 81 00 00    00 00 24 00    24 00 1B 01
  40:0020 :    1B 01 1B 01    00 00 00 00    00 00 00 00    00 00 00 00
  40:0030 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 01 01
  40:0040 :    25 00 04 00    00 2B 01 0F    02 03 50 00    00 10 00 00
  40:0050 :    00 18 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:0060 :    0E 0D 00 D4    03 29 30 C2    11 85 76 00    A1 C9 08 00
  40:0070 :    00 00 00 12    00 02 00 00    14 14 14 14    01 01 01 01
  40:0080 :    1E 00 3E 00    18 10 00 60    F9 11 0B 01    00 00 00 05
  40:0090 :    17 00 00 00    2B 00 10 00    00 00 00 00    00 00 00 00
  40:00A0 :    00 00 00 00    00 00 00 00    7C 14 00 C0    00 00 00 00
  40:00B0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00C0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00D0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00E0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00
  40:00F0 :    00 00 00 00    00 00 00 00    00 00 00 00    00 00 00 00


Interrupt Vector Table (including INSPECT)
  00 - 03 :    0A6A:0555      0070:0465      07C6:0016      0070:0465
  04 - 07 :    0070:0465      F000:FF54      F000:93CC      F000:9BD0
  08 - 0B :    07C6:001F      07C6:0028      F000:9BD0      07C6:0052
  0C - 0F :    F000:9BD0      F000:9BD0      07C6:009A      0070:0465
  10 - 13 :    C000:13FE      F000:F84D      F000:F841      0070:03EE
  14 - 17 :    F000:F66C      0206:0240      0070:042D      F000:EFD2
  18 - 1B :    F000:9C93      0887:002F      F000:FE6E      0070:045F
  1C - 1F :    F000:FF53      F000:0000      0000:0522      C000:2143
  20 - 23 :    00C9:0FA8      00C9:0FB2      088F:0314      088F:016D
  24 - 27 :    088F:0178      00C9:0FBC      00C9:0FC6      00C9:0FD0
  28 - 2B :    00C9:106F      0070:0466      00C9:106F      00C9:106F
  2C - 2F :    00C9:106F      00C9:106F      088F:0162      0890:01CC
  30 - 33 :    C90F:E4EA      F000:9B00      00C9:106F      00C9:106F
  34 - 37 :    00C9:106F      00C9:106F      00C9:106F      00C9:106F
  38 - 3B :    00C9:106F      00C9:106F      00C9:106F      00C9:106F
  3C - 3F :    00C9:106F      00C9:106F      00C9:106F      1B26:04F3
  40 - 43 :    F000:EC59      C81F:01C6      F000:F065      C000:2556
  44 - 47 :    F000:9BD0      F000:9BD0      C81F:01E0      F000:9BD0
  48 - 4B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  4C - 4F :    F000:9BD0      F000:9BD0      F000:9BD0      0070:04FC
  50 - 53 :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  54 - 57 :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  58 - 5B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  5C - 5F :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  60 - 63 :    0000:0000      0000:0000      0000:0000      0000:0000
  64 - 67 :    0000:0000      0000:0000      0000:0000      0000:0000
  68 - 6B :    F000:9BD0      F000:9BD0      F000:9BD0      F000:9BD0
  6C - 6F :    F000:9BD0      C000:13FE      F000:9BD0      F000:9BD0
  70 - 73 :    07C6:0035      F000:9C1F      F000:9BD0      F000:9BD0
  74 - 77 :    07C6:00E2      F000:9C28      07C6:00FA      F000:9BD0
  78 - 7B :    0000:0000      0000:0000      0000:0000      0000:0000
  7C - 7F :    0000:0000      0000:0000      0000:0000      0000:0000
  80 - 83 :    0000:0000      0000:0000      0000:0000      0000:0000
  84 - 87 :    0000:0000      0000:0000      0000:0000      0000:0000
  88 - 8B :    0000:0000      0000:0000      0000:0000      0000:0000
  8C - 8F :    0000:0000      0000:0000      0000:0000      0000:0000
  90 - 93 :    0000:0000      0000:0000      0000:0000      0000:0000
  94 - 97 :    0000:0000      0000:0000      0000:0000      0000:0000
  98 - 9B :    0000:0000      0000:0000      0000:0000      0000:0000
  9C - 9F :    0000:0000      0000:0000      0000:0000      0000:0000
  A0 - A3 :    0000:0000      0000:0000      0000:0000      0000:0000
  A4 - A7 :    0000:0000      0000:0000      0000:0000      0000:0000
  A8 - AB :    0000:0000      0000:0000      0000:0000      0000:0000
  AC - AF :    0000:0000      0000:0000      0000:0000      0000:0000
  B0 - B3 :    0000:0000      0000:0000      0000:0000      0000:0000
  B4 - B7 :    0000:0000      0000:0000      0000:0000      0000:0000
  B8 - BB :    0000:0000      0000:0000      0000:0000      0000:0000
  BC - BF :    0000:0000      0000:0000      0000:0000      0000:0000
  C0 - C3 :    0000:0000      0001:0000      0000:0000      0000:0000
  C4 - C7 :    0000:0000      0000:0000      0000:0000      0000:0000
  C8 - CB :    0000:0000      0000:0000      0000:0000      0000:0000
  CC - CF :    0000:0000      0000:0000      0000:0000      0000:0000
  D0 - D3 :    0000:0000      0000:0000      0000:0000      0000:0000
  D4 - D7 :    0000:0000      0000:0000      0000:0000      0000:0000
  D8 - DB :    0000:0000      0000:0000      0000:0000      0000:0000
  DC - DF :    0000:0000      0000:0000      0000:0000      0000:0000
  E0 - E3 :    0000:0000      0000:0000      0000:0000      0000:0000
  E4 - E7 :    0000:0000      0000:0000      0000:0000      0083:0000
  E8 - EB :    0002:00D0      0083:0018      0002:00D8      0006:00D8
  EC - EF :    0046:8500      1FA5:00F6      0046:0087      13C1:00F4
  F0 - F3 :    7C00:0009      D53F:1DA6      1DA6:13C1      1400:D39C
  F4 - F7 :    1CDA:0246      0101:7387      0000:0000      0000:613D
  F8 - FB :    613D:0020      15B7:6443      0000:0003      0286:09C0
  FC - FF :    0000:00F4      0000:09C0      4E74:0049      0003:09E5

PCI Devices Information
  Signature  . . . . . .  PCI
  Config Mechanism #1  . . . . .  Supported
  Config Mechanism #2  . . . . .  Not Supported
  Spec Cycle for Config #1 . . .  Supported
  Spec Cycle for Config #2 . . .  Not Supported
  BIOS Interface Version . . . .  2.10
  Last PCI Bus Number  . . . . .  10
  Number of PCI Devices  . . . .  6

  PCI Bus Number . . . . . .  0
  Device Number  . . . . . .  4
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  0
  Vendor ID  . . . . . .  0E11h
  Device ID  . . . . . .  0010h
  Revision ID  . . . . . .  02h
  Device Type  . . . . . .  RAID Controller
  Programming Interface  . . . .  00h
  Expansion ROM Base Address . .  FFF80000h
  IRQ Line . . . . . .  3
  IRQ Pin  . . . . . .  INTA#
  IO Address Base  . . . . . .  2000h
  IO Address Length  . . . . . .  100h
  Memory Address Base  . . . . .  EC000000h
  Memory Address Length  . . . .  1000000h
  Memory Address Base  . . . . .  EB000000h
  Memory Address Length  . . . .  1000000h

  PCI Bus Number . . . . . .  0
  Device Number  . . . . . .  5
  Function Number  . . . . . .  00h
  Slot Number  . . . . . .  0
  Vendor ID  . . . . . .  1002h
  Device ID  . . . . . .  4756h
```

```
Revision ID  . . . . . .  7Ah
Device Type  . . . . . .  VGA Compatible Controller
Programming Interface  . . . .  00h
Expansion ROM Base Address . .  FFFE0000h
IRQ Line . . . . . .  255
IRQ Pin  . . . . . .  Not Used
Memory Address Base  . . . . .  E9000000h
Memory Address Length  . . . .  1000000h
IO Address Base  . . . . . .  2400h
IO Address Length  . . . . . .  100h
Memory Address Base  . . . . .  EAFF0000h
Memory Address Length  . . . .  1000h

PCI Bus Number . . . . . .  0
Device Number  . . . . . .  7
Function Number  . . . . . .  00h
Slot Number  . . . . . .  6
Vendor ID  . . . . . .  0E11h
Device ID  . . . . . .  B060h
Revision ID  . . . . . .  02h
Device Type  . . . . . .  RAID Controller
Programming Interface  . . . .  00h
Expansion ROM Base Address . .  FFF00000h
IRQ Line . . . . . .  5
IRQ Pin  . . . . . .  INTA#
Memory Address Base  . . . . .  EAF80000h
Memory Address Length  . . . .  40000h
Memory Address Base  . . . . .  EAE00000h
Memory Address Length  . . . .  100000h
IO Address Base  . . . . . .  2800h
IO Address Length  . . . . . .  100h

PCI Bus Number . . . . . .  2
Device Number  . . . . . .  6
Function Number  . . . . . .  00h
Slot Number  . . . . . .  1
Vendor ID  . . . . . .  0E11h
Device ID  . . . . . .  1229h
Revision ID  . . . . . .  08h
Device Type  . . . . . .  Ethernet Controller
Programming Interface  . . . .  00h
Expansion ROM Base Address . .  FFF00000h
IRQ Line . . . . . .  10
IRQ Pin  . . . . . .  INTA#
Memory Address Base  . . . . .  F2DE0000h
Memory Address Length  . . . .  1000h
IO Address Base  . . . . . .  3000h
IO Address Length  . . . . . .  40h
Memory Address Base  . . . . .  F2C00000h
Memory Address Length  . . . .  100000h

PCI Bus Number . . . . . .  2
Device Number  . . . . . .  8
Function Number  . . . . . .  00h
Slot Number  . . . . . .  2
Vendor ID  . . . . . .  0E11h
Device ID  . . . . . .  B060h
Revision ID  . . . . . .  02h
Device Type  . . . . . .  RAID Controller
Programming Interface  . . . .  00h
Expansion ROM Base Address . .  FFF00000h
IRQ Line . . . . . .  11
IRQ Pin  . . . . . .  INTA#
Memory Address Base  . . . . .  F2BC0000h
Memory Address Length  . . . .  40000h
Memory Address Base  . . . . .  F2A00000h
Memory Address Length  . . . .  100000h
IO Address Base  . . . . . .  3400h
IO Address Length  . . . . . .  100h

PCI Bus Number . . . . . .  7
Device Number  . . . . . .  7
Function Number  . . . . . .  00h
Slot Number  . . . . . .  5
Vendor ID  . . . . . .  0E11h
Device ID  . . . . . .  B060h
Revision ID  . . . . . .  02h
Device Type  . . . . . .  RAID Controller
Programming Interface  . . . .  00h
Expansion ROM Base Address . .  FFF00000h
IRQ Line . . . . . .  11
IRQ Pin  . . . . . .  INTA#
Memory Address Base  . . . . .  F7F80000h
Memory Address Length  . . . .  40000h
Memory Address Base  . . . . .  F7E00000h
Memory Address Length  . . . .  100000h
IO Address Base  . . . . . .  5000h
IO Address Length  . . . . . .  100h
```

ProLiant ML570 is a trademark of Compaq Computer Corporation.

Microsoft SQL Server 2000 Installation Procedures

```
Microsoft SQL Server 2000 Installation Procedures
Type of installation: custom
During the custom installation, use the default
settings for all except the following two areas:
Services accounts:
SQL Server - local system account
SQL Server Agent - local system account
Set the sort order/collation as binary sort order/Latin_1_General
```

Microsoft COM Component Configuration Parameters

```
The component services tool in Windows 2000
was used to change the queue settings for the
TPCC COM+ single queue component. The
single queue component was set to enable
object pooling, object construction, just in time
activation, and component supports events and
statistics. The min and max pool size for the
single queue component on each client was 105.
Delivery threads were set under the TPCC key
in the registry. The construction string was Server =
myserver; UID= sa; pwd=; DATABASE= tpcc;
```

Appendix D:
180-Day Space

Per TPC-C Version 5 upgrade rules, the 180-day space exceeds the TPC-C Version 5 – 60 Day Space requirement.

TPC-C 180 Day Space Requirements

Warehouses	1620				TpmC	20,207.20
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	1,620	176	32	10		218
District	16,200	1,800	32	92		1924
Customer	48,600,000	35,345,456	2,107,680	1,872,657		39325793
History	48,600,000	2,700,008	48		567,694	2700056
NewOrder	14,580,000	230,520	544	11,553		242617
Orders	48,600,000	1,489,656	677,416		1,185,998	2167072
OrderLine	485,997,840	30,374,840	64,304		6,066,735	30439144
Item	100,000	9,528	48	479		10055
Stock	162,000,000	51,840,000	96,928	2,596,846		54533774
Total		121,991,984	2,947,032	4,481,637	7,820,427	129,420,653

MB		
Dynamic Space	33,754	Sum of Data for Order, Orderline and History
Static Space	92,633	Sum of Data+Index+5%-Dynamic Space
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)
Daily Growth	6,737	(Dynamic Space/(W*62.5))*tpmc
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed

180 Day Space MB	1,305,223					
180 Day Space GB	1,274.63	GB				
Log Size	51,999.99	MB				
KB Per New Order	5.22	KB				
8 hr log MB	49,445	MB				
8 hr log GB	48.2859	GB				
Space Usage	GB Needed	Measured	GB Priced	Disk Size	Formatted Size	
180 Day Space DB	1,274.63	78	1349.40	18GB	17.300	
		0	0.00	9GB	8.473	
			0.00	4GB	3.999	
Total DB		78.00	1349.40	9GB		
8-hr log + mirror	96.5719	6	103.80	18GB	8.473	
OS, Swap	3	1	8.473	9GB		
Total Storage	1,374.20	GB	1,461.67	GB		

tpmC	21,000.00

	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	KB/New-Order	8-Hr Growth KB	8-Hr Growth MB
History	2,730,152	96	2,809,928	96	79,776	0.0585	589,966.96	576.14
Order	1,619,256	1,286,456	1,640,656	1,359,088	94,032	0.1223	1,232,529.26	1,203.64
Order-Line	30,919,920	128,592	31,772,456	128,592	852,536	0.6255	6,304,754.25	6,156.99
								7,936.77

	sum(*) Before
d_next_o_id	49,130,610

	sum(*) After	Num New-
	50,493,639	1,363,029

	Before MB	After MB	Grow MB	KB/New-Order	8-Hr Growth MB	8-Hr Growth GB
Log	3991.10	10939.40	6948.30	5.2200	51,384.70	50.18
				5.345.3138	bytes	

51999.99219	7.675196		21.037308
Database tpcc log used (%)			

Appendix E: Third Party Letters

One Microsoft Way
Redmond, WA 98052-6399
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

August 17, 2000

Mr. Brean Campbell
Compaq Computer Corp.
P.O. Box 692000
Houston, TX 77269-2000

Dear Mr. Campbell:

Here is the information you requested regarding U.S. pricing for several Microsoft products, to be used in conjunction with TPC-C benchmark testing.

Part Number	Description	Unit Price	Quantity	Price
228-00968	**SQL Server 2000 Standard Edition** Per processor licensing Discount schedule: Open Program Level B	$4,271	3	$12,813
C11-00821	**Windows 2000 Server** Server license only – No CALs Discount schedule: Open Program – No Level	$738	1	$738
048-00317	**Visual C++ Professional 6.0 Win32**	$549	1	$549
	5-year maintenance for above software	$2,095	5	$10,475

This quote is valid for the next 90 days.

If I can be of any further assistance, please contact me at (425) 705-9857 or kurtdan@microsoft.com.

Yours truly,

Kurt Daniel
Business Manager
SQL Server Marketing

Home : Go Back : **Product Detail**

Product Detail

	Plat.	Item #	Manufacturer/Product	Availability	Price
ADD TO CART	PC	105692	LINKSYS GROUP 10 100BaseTX EtherFast 8-Port Rackmount Switch	Today	$139.95

The EtherFast 10/100 Switch is the easiest, most flexible way to boost your network's performance while migrating to the power of Fast Ethernet.

Designed to meet the needs of small businesses, corporate workgroups and SOHO networks, the (8) port EtherFast Switch from Linksys has support for half and full duplex speeds, allowing your network to run at 10Mbps, 20Mbps, 100Mbps and an incredible 200Mbps in full duplex mode! EtherFast Switches eliminate bottlenecks and bandwidth constraints, optimizing network performance while protecting your investment in your existing infrastructure. It has (8) RJ-45 ports for 10/100BaseTX connection. One of the ports has a shared uplink switch for daisy-chaining to another hub or network backbone.

The EtherFast 10/100 Switches pack a full suite of error detection and correction features for reliable communication every time. Auto-partitioning, data collision control, preamble regeneration and incoming frame retiming ensures that not a single bit is lost, even during the heaviest moments of network activity. Built to last, these switches are optimized to deliver high-end video, multimedia and other speed-intensive applications at blistering speeds.

The unit is IEEE 802.3 and 802.3u 10/100BaseTX Class II Compliant and has a small Footprint - perfect for small and medium workgroups and Ethernet Frame types. It is fully compliant with all major network Operating Systems and comes with free technical support and a five year warranty.

Product Features

# of Ports	8
Managed (SNMP Support)	No
Wiring Type(s) Supported	100BaseTX (Twisted Pair)
Wiring Type(s) Supported	10BaseT (Twisted Pair)
Protocol(s) Supported	Ethernet/Fast Ethernet
Form Factor	External (Desktop/stackable)
Speed	10/100 Mbps
Dedicated Cascade Port	No

Compatiblity Notes

IEEE 802.3 and 802.3u standards for Ethernet/Fast Ethernet 10/100BaseTX

CSMA/CD

Ports/Connectors

(8) RJ-45 ports for 10/100BaseTX connection

Product Details

Money Back Guarantee	N/A
Labor Warranty	5 Years
Product Warranty	5 Years
Shipping Weight	6.5 lbs
Mfr. Part No.	EZXS88R