

TPC Benchmark™ C
Full Disclosure Report
for
Lenovo SureServer T350
using
Microsoft SQL Server 2005 Enterprise x64 Edition
and
Windows Server 2003 Enterprise x64 Edition SP1

TPC-C Version 5.6

Submitted for Review
May 1, 2006

lenovo

First Edition, May 1, 2006

Lenovo believes that the information included in this document is accurate as of the publication date. The information in this document is subject to change without notice. Furthermore, Lenovo is not responsible for any errors contained within this document. The pricing information given in this FDR is accurate as of the publication date, May 1 2006. But Lenovo can not guarantee that all sources will offer the same pricing.

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

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

Lenovo and SureServer are registered trademarks of Lenovo.

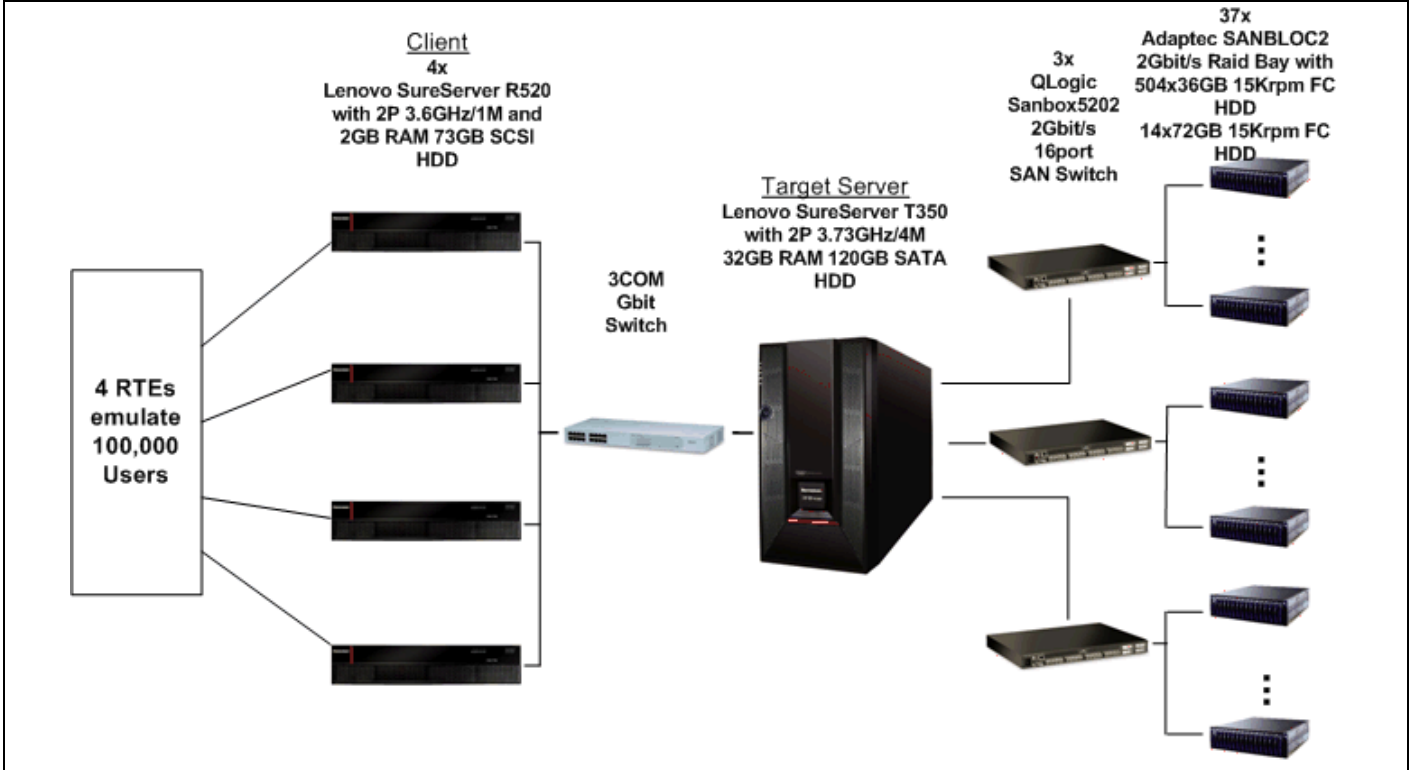
Microsoft, Windows 2003, SQL Server and COM+ are either trademarks or registered trademarks of Microsoft Corporation.

Intel and Intel Xeon are registered trademarks of Intel Corporation.


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

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

lenovo		SureServer T350		TPC-C Rev. 5.6	
		C/S with 4 R520		Report Date: May 1, 2006	
Total System Cost		TPC-C Throughput	Price/Performance		Availability Date
8,265,969 CNY		125,954	65.63		July 1, 2006
		CNY/tpmC			
Database Server Processors /Cores/Threads	Database Manager	Operating System	Other Software	Number of Users	
2/4/8 Intel® Xeon® processor 3.73GHz Dual Core (4MB L2 Cache)	Microsoft SQL Server 2005, Enterprise x64 Edition	Microsoft Windows Server 2003, Enterprise X64 Edition, SP1	Microsoft Windows 2003 Standard Edition, SP1 Microsoft Visual C++ Microsoft COM+	100,000	



System Components	Qty	Server	Qty	Each client
Processors/Cores/Threads	2/4/8	Intel® Xeon® Processor 3.73GHz 4MB L2 Cache	2/2/4	Intel® Xeon® Processor 3.6GHz 1MB L2
Memory	8	4GB DDR2 533 FBD	2	1GB DDR2 400
Disk Controllers	2	QLogic QLA2462		
Disk Drives	1	120GB 7200rpm SATA150	1	73GB 10Krpm SCSI
	504	36GB 15Krpm FC		
	14	72GB 15krpm FC		
Total Storage		19272GB		73GB
Others				

		SureServer T350 Client/Server			TPC-C Rev 5.6		
				Report Date: 1-May-2006			
Description	Part Number	Third Party	Unit Price CNY	Qty	Extended Price CNY	3 yr. Maint. Price CNY	
Server Hardware							
Lenovo SureServer T350 2P Xeon DC 3.73GHz 32GB DDR2 533 FBD 1 x 120GB SATA150 7200rpm HDD, keyboard, mouse	61-012654	Lenovo	1	308,040.00	1	308040.00	6100.00
Lenovo monitor(17" CRT monitor)	88000376	Lenovo	1	891.00	1	891.00	0.00
APC Smart UPS 1500 rackmount	SUA1500RM12U	APC	2	3,450.00	1	3450.00	1300.00
APC Smart UPS 1500 rackmount (spare)	SUA1500RM12U	APC	2	3,450.00	2	6900.00	2600.00
Subtotal					319281.00	10000.00	
3rd Party FC HBA							
Qlogic QL2462 dual port fibre channel host bus adapter	QLE2462-SP	Qlogic	1	15,405.60	2	30811.20	3389.23
Subtotal					30811.20	3389.23	
Storage Subsystem							
Adaptec Sanbloc2 dual raid storage subsystem	FC2501DR2-AC	Adaptec	1	174,919.71	1	174919.71	0.00
Adaptec Sanbloc2 single raid storage subsystem	FC2501SR2-AC	Adaptec	1	102,889.71	36	3704029.56	0.00
5 Meter 2Gb Fibre Channel Optical Cable, Multi Mode, LC to LC	2129000	Adaptec	1	926.10	41	37970.10	0.00
72 GB 15,000 rpm Fibre Channel Drive in Carrier (Seagate Cheetah X15)	2149100	Adaptec	1	5,762.40	14	80673.60	0.00
36 GB 15,000 rpm Fibre Channel Drive in Carrier (Seagate Cheetah X15)	2149400	Adaptec	1	3,910.20	504	1970740.80	0.00
Lenovo 19" 42U Enclosure	30000853	Lenovo	1	8,733.00	5	43665.00	4803.15
Lenovo KVM Switch & Cable (8 port)	888004199	Lenovo	1	2,381.50	1	2381.50	0.00
10% spare 36 GB 15,000 rpm Fibre Channel Drive	2149400	Adaptec	1	3,910.20	51	199420.20	0.00
10% spare 72 GB 15,000 rpm Fibre Channel Drive	2149100	Adaptec	1	5,762.40	2	11524.80	0.00
10% spare Adaptec Sanbloc2 dual raid storage subsystem	FC2501DR2-AC	Adaptec	1	174,919.71	2	349839.42	0.00
10% spare Adaptec Sanbloc2 single raid storage subsystem	FC2501SR2-AC	Adaptec	1	102,889.71	4	411558.84	0.00
5 Meter 2Gb Fibre Channel Optical Cable, Multi Mode, LC to LC (10% spare)	2129000	Adaptec	1	926.10	5	4630.50	0.00
Subtotal					6991354.03	4803.15	
Server Software							
Microsoft SQL Server 2005 Enterprise x64 Edition (per processor)	810-03152	Microsoft	1	215,000.00	2	430000.00	
Microsoft Windows Server 2003, Enterprise x64 Edition, SP1	38007772	Microsoft	1	25,000.00	1	25000.00	
Professional Support (1 Incident)		Microsoft	1	1,000.00	1		1000.00
Subtotal					455000.00	1000.00	
Client Hardware							
Lenovo SureSever R520 2 x Xeon 3.6 GHz w/1MB L2 cache, 2 GB DDR2-400 RAM (2 x 1GB), 73GB HDD 10K RPM, keyboard, mouse	61-012653	Lenovo	1	33,728.40	4	134913.60	24400.00
Lenovo Monitor (17" CRT monitor)	88000376	Lenovo	1	891.00	4	3564.00	0.00
Subtotal					138477.60	24400.00	
Client Software							
Microsoft Windows 2003 Server, Standard x86 Edition, SP1	38007760	Microsoft	1	4,500.00	4	18000.00	
Visual C++ .Net Standard (Visual Studio Pro 2005)	C5E-00116	Microsoft	1	4,000.00	1	4000.00	
Subtotal					22000.00	0.00	
Connectivity							
3COM SuperStack 3 Switch 4400 PWR 24P	3C17205	3COM	3	16,484.00	1	16484.00	0.00
3COM SuperStack 3 Switch 4400 PWR 24P (spare)	3C17205	3COM	3	16,484.00	2	32968.00	0.00
Qlogic Sanbox 5202 16port	SB5202-16A	Qlogic	1	65,000.00	3	195000.00	21000.00
Subtotal					244452.00	21000.00	
Total					8201375.83	64592.38	
Pricing key:1. Lenovo, 2. Huiyuan, (86)-10-62610081, http://detail.zol.com.cn/61/60869/price.shtml , 3. Skylevel, (86)-20-85614920, http://www.skylevel.com.cn/baojia(3com).htm				Three-Year Cost of Ownership		CNY 8,265,969	
Audited by Lorna Livingtree of Performance Metrics Inc.				tpmC Rating		125,954	
				Cost/tpmC		CNY 65.63	
Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these items, please inform the TPC at pricing@tpc.org . Thank you.							

Numerical Quantities Summary			
MQTh, Computed Maximum Qualified Throughput: 125,954 tpmC			
Response Times(in seconds)	90th Percentile	Average	Maximum
New-Order	0.22	0.17	5.09
Payment	0.17	0.13	2.44
Delivery	0.11	0.10	1.09
Stock Level	0.78	0.61	3.41
Order Status	0.19	0.16	5.04
Delivery (Deferred)	0.13	0.10	5.00
Menu	0.11	0.10	1.51
Response Time Delay Added for Emulated Components: (sec)			menu 0.1 Resp 0.1
Transaction Mix (in percent of total transactions)			Percent
New-Order			44.74%
Payment			43.06%
Delivery			4.04%
Stock Level			4.08%
Order Status			4.08%
Keying/Think Times(in seconds)	Minimum	Average	Maximum
New-Order	18.02/0.00	18.04/12.07	18.55/121.01
Payment	3.02/0.00	3.03/12.06	3.55/121.01
Delivery	2.02/0.00	2.03/5.08	2.55/50.52
Stock Level	2.02/0.00	2.03/5.06	2.55/50.52
Order Status	2.02/0.00	2.03/10.05	2.54/100.53
Test Duration			
Ramp-up time			146minutes
Measurement interval			120minutes
Number of checkpoints			4
Checkpoint interval			29min30sec
Number of transactions (all types) completed in measurement interval			87,533,323

Abstract.....	10
Preface	11
General Items	12
Test Sponsor.....	12
Application Code and Definition Statements.....	12
Parameter Settings.....	12
Configuration Items	12
Clause 1 Related Items.....	14
Table Definitions.....	14
Physical Organization of Database	14
Priced Configuration vs. Measured Configuration:	14
Insert and Delete Operations.....	14
Horizontal or Vertical Partitioning.....	14
Replication	14
Table Attributes	14
Clause 2: Transaction and Terminal Profiles Related Items.....	15
Random Number Generation	15
Screen Layout	15
Terminal Verification	15
Intelligent Terminals	15
Transaction Profiles	15
Deferred Delivery Mechanism.....	16
Clause 3 Related Items.....	17
Transaction System Properties (ACID).....	17
Atomicity	17
Consistency	17
Isolation.....	17
Durability	18
Clause 4 Related Items.....	19
Initial Cardinality of Tables	19
Database Layout.....	19
Type of Database.....	22
Database Mapping.....	22
60 Day Space	22
Clause 5 Related Items.....	23
Throughput.....	23
Response Times	23
Keying/Think Times	23
Response Time Frequency Distribution Curves.....	24
Performance Curve for Response Time vs. Throughput	26
Work Performed During Steady State	28
Measurement Period Duration	28
Regulation of Transaction Mix.....	28
Transaction Statistics.....	29
Checkpoint Count and Location.....	29
Checkpoint Duration	29
Clause 6 Related Items.....	31
RTE Descriptions	31
Emulated Components.....	31
Functional Diagrams	31
Networks.....	31
Operator Intervention.....	31

Clause 7 Related Items.....	32
System Pricing	32
Availability, Throughput, and Price Performance.....	32
Country Specific Pricing.....	32
Usage Pricing.....	32
Clause 8 Audit Related Items.....	33
Auditor’s Report	33
Availability of the Full Disclosure Report	33
Attestation Letter.....	33
Appendix A: SOURCE CODE	36
Web Client Source Code	36
error.h	36
ReadRegistry.cpp	43
ReadRegistry.h	46
trans.h.....	47
txn_base.h	51
rtetime.h	52
spinlock.h.....	52
txnlog.h	55
tpcc_dblib.cpp.....	64
tpcc_dblib.h.....	85
tpcc_odbc.cpp	89
tpcc_odbc.h.....	112
tpcc_oledb.cpp	116
tpcc_oledb.h.....	156
install.c	161
install.h.....	182
install.rc.....	183
install_com.cpp	189
RESOURCE.H.....	195
resource.h	196
tpcc.cpp	197
tpcc.def.....	245
tpcc.h.....	245
tpcc.rc.....	250
tpcc_com.cpp	253
tpcc_com.h	257
client_utils.c	259
client_utils.h.....	263
mon_client.c.....	265
mon_client.h.....	272
tpcc_enc.cpp.....	273
tpcc_enc.h	276
tpcc_tux.cpp.....	278
tpcc_tux.h.....	283
Methods.h.....	285
resource.h	290
tpcc_com_all.cpp	291
tpcc_com_all.def.....	302
tpcc_com_all.h.....	302
tpcc_com_all.idl.....	305
tpcc_com_all.rc.....	307
tpcc_com_all.rgs	310

tpcc_com_all_i.c	310
tpcc_com_no.rgs	314
tpcc_com_os.rgs	315
tpcc_com_pay.rgs	315
tpcc_com_ps.h	316
tpcc_com_sl.rgs	321
dlldata.c	322
tpcc_com_ps.def	323
tpcc_com_ps.h	323
tpcc_com_ps.idl	329
tpcc_com_ps_i.c	330
tpcc_com_ps_p.c	333
tuxapp.cpp	376
tuxapp.h	384
tuxmain.c	385
Webclnt.sln	387
Stored Procedures	389
delivery.sql	389
NewOrd.sql	392
null-txns.sql	398
ordstat.sql	405
payment.sql	408
StockLev.sql	412
tpcc_neworder_new.sql	413
version.sql	417
Appendix B: DATABASE DESIGN	419
Database Build	419
backup.sql	419
backupdev.sql	419
createdb.sql	420
dropdev.sql	424
removedb.sql	424
restore.sql	425
idxcuscl.sql	425
idxcusnc.sql	426
idxdiscl.sql	427
idxhiscl.sql	427
idxitmcl.sql	428
idxnodcl.sql	429
idxodlcl.sql	429
idxordcl.sql	430
idxordnc.sql	431
idxstkcl.sql	431
idxwarcl.sql	432
tables.sql	433
Load Source Code	436
dbopt1.sql	436
dbopt2.sql	437
RunSQLCfg.sql	439
sqlshutdown.sql	440
VerifyTPCCLoad.sql	440
version.sql	447
Appendix C: TUNABLE PARAMETERS	449

Microsoft SQL Server 2005 Configuration Parameters	449
Microsoft Windows Server 2003 Enterprise Edition SP1	451
Client Configuration Parameters.....	539
Microsoft Windows Server 2003 Standard Edition SP1 Client System Information Report	539
Microsoft Windows Server 2003 Standard Edition SP1 Client Registry Parameters	591
Microsoft SQL Server 2005 installation procedure.....	592
Microsoft COM+ component configuration parameters	593
Benchcraft Profile	593
Appendix D: 60-DAY SPACE	599
APPENDIX E: Third Party Letters	600

Abstract

Lenovo conducted the TPC Benchmark™ C on the SureServer T350 configured as a client/server system. This report documents the full disclosure information required by the TPC Benchmark Standard Specification, Revision 5.6, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

The software used on the SuerServer T350 system includes Microsoft® Windows® Server 2003 Enterprise x64 Edition operating system and Microsoft SQL Server 2005 Enterprise Edition database. Two standard metrics, transactions per minute-C (tpmC) and price per tpmC , are reported as required by the TPC Benchmark C Standard Specification.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	tpmC	\$/tpmC	Total Solution Availability Date
SureServer T350	Microsoft SQL Server 2005 Enterprise x64 Edition Microsoft Windows Server 2003 Enterprise x64 Edition, SP1	CNY 8,265,969	125,954	65.63 CNY/tpmC	July 1, 2006

The results of the benchmark and test methodology used were audited by Lorna Livingtree of Performance Metrics, Inc. to validate compliance with the TPC specifications.

Preface

The TPC Benchmark™ C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specification Version 5.6.

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

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

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

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

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

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

General Items

Test Sponsor

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

This benchmark was sponsored by Lenovo.

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: SOURCE CODE 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.

Error! Reference source not found. contains the tunable parameters to for the database, the operating system, and the transaction monitor.

Configuration Items

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

The configuration diagram for both the measured and priced systems are shown as **Error! Reference source not found.** as Figure 2.

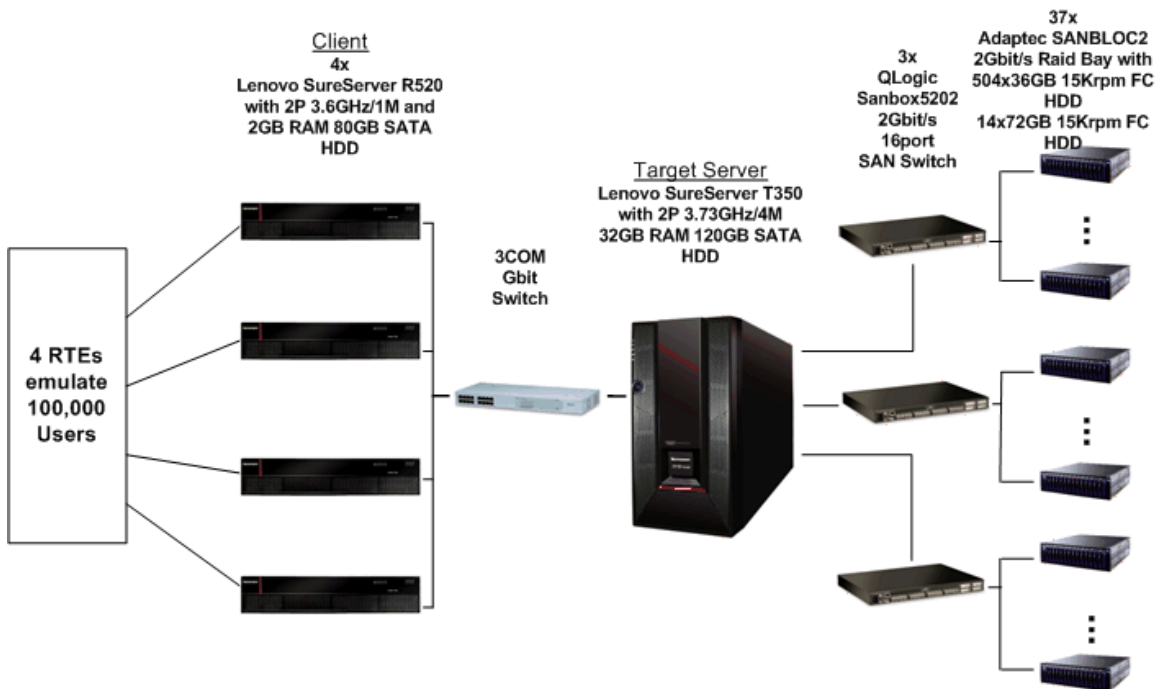


Figure 1 Measured Configuration

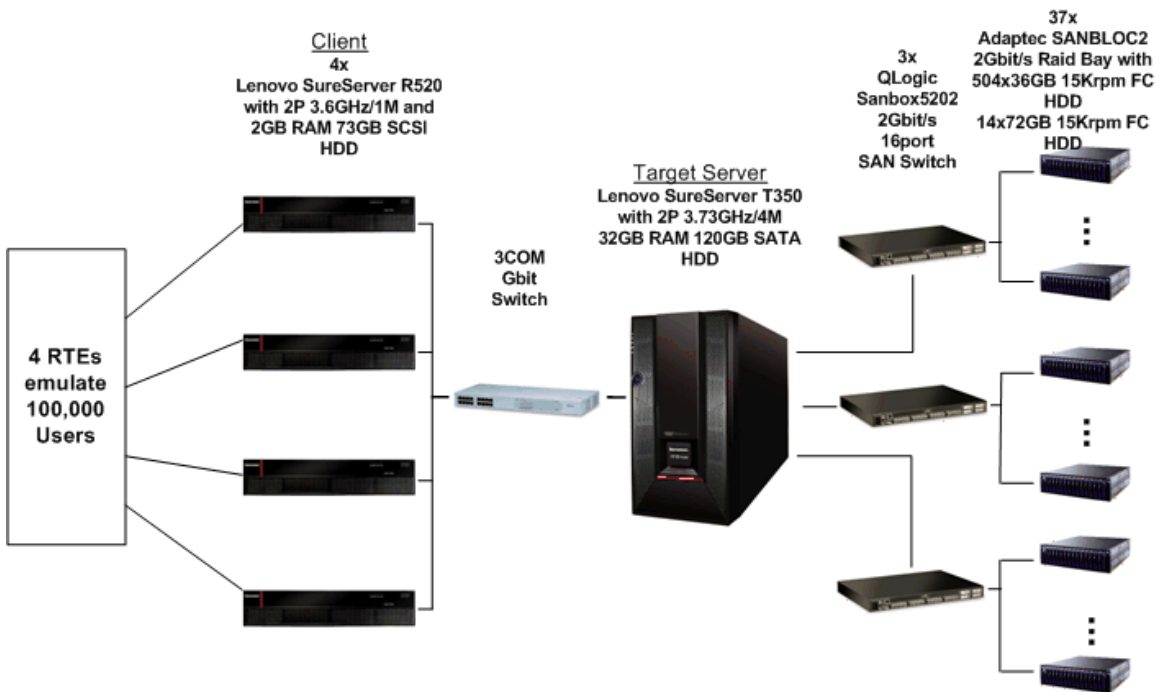


Figure 2 Priced Configuration

Clause 1 Related Items

Table Definitions

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

Error! Reference source not found. 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.

Physical space was allocated to Microsoft SQL Server on the server disks as detailed in Table 3.

Priced Configuration vs. Measured Configuration:

The configuration diagram for both the measured and priced systems are shown as **Error! Reference source not found.** as Figure 2.

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 running of the benchmark. The space required for an additional 5 percent of the initial table cardinality was allocated to Microsoft SQL Server 2005 and priced as static space.

Horizontal or Vertical Partitioning

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

Partitioning was not used in this benchmark.

Replication

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

Replication was not used in this benchmark.

Table Attributes

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

No additional attributes were used in this benchmark.

Clause 2: Transaction and Terminal Profiles Related Items

Random Number Generation

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

The random number generation was done internal to the Microsoft BenchCraft RTE program, which was audited independently.

Screen Layout

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

All screen layouts followed the TPC Benchmark C Standard Specification.

Terminal Verification

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

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

Intelligent Terminals

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: SOURCE CODE.

Transaction Profiles

*The percentage of home and remote order-lines in the New-Order transactions must be disclosed.
The percentage of New-Order transactions that were rolled back as a result of an unused item number must be disclosed.*

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

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

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

Table 1 Transaction Statistics

New Order	Value
Home warehouse order lines	99.00%
Remote warehouse order lines	1.00%
Rolled back transactions	1.00%
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	84.99%
Remote warehouse payment transactions	15.01%
Non-Primary Key Access	
Payment transactions using C_LAST	60.01%
Order-Status transactions using C_LAST	60.02%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.74%
Payment	43.06%
Delivery	4.04%
Stock Level	4.08%
Order Status	4.08%

Deferred Delivery Mechanism

The queuing mechanism used to defer 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: SOURCE CODE.

Clause 3 Related Items

Transaction System Properties (ACID)

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

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

Atomicity

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

Completed Transactions

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

Aborted Transactions

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

Consistency

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

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

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

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

Isolation

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

Isolation tests one through 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.

Loss of Data and Log

This test was executed on a fully scaled database of 10,000 warehouses under a full load of 100,000 users, which accessed only the first 1500 warehouses. To demonstrate recovery from a permanent failure of durable medium containing DBMS logs and TPC-C tables, the following steps were executed:

1. 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.
2. The RTEs were started with 15,000 users.
3. The test was allowed to run for a minimum of 10 minutes.
4. One log disk was removed from the server.
5. Since the disk was mirrored, processing was not interrupted. This was verified by checking the user status on the RTE.
6. One of the data disks was removed from a Adaptec Sanbloc2 Single Raid Storage Subsystem.
7. When Microsoft SQL Server recorded errors about not being able to access the database, the RTE was shut down and SQL stopped.
8. Microsoft SQL Server was restarted. A dump of the transaction log was taken.
9. SUT server was shutdown.
10. Both disks were replaced with new disks, and the log drive was allowed to complete RAID1 recovery.
11. SUT server was rebooted and data drive is brought online.
12. The database was restored from backup and the transaction log dump was applied.
13. Step 1 was repeated and the difference between the first and second counts was noted.
14. An RTE report was generated for the entire run time giving the number of NEWORDERS successfully returned to the RTE.
15. It was verified that the sum of D_NEXT_O_ID after the database is recovered is greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.
16. Consistency condition #3 was executed and verified.
17. 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 10,000 warehouses under a full load of 100,000 users. The following steps were executed:

1. 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.
2. The RTE was started with 100,000 users.
3. The test was allowed to run for a minimum of 10 minutes.
4. System crash and loss of memory were induced by pulling the power cord out of the system.
5. The RTE was shutdown.
6. Power was restored and the system restarted.
7. Microsoft SQL Server was restarted and performed an automatic recovery.
8. Step 1 was repeated and the difference between the first and second counts was noted.
9. Consistency condition #3 was executed and verified.

10. An RTE report was generated for the entire run time giving the number of NEWORDERS successfully returned to the RTE.
11. It was verified that the sum of D_NEXT_O_ID after the database recovered was greater than or equal to the sum of D_NEXT_O_ID before the run, plus all new order transactions completed during the run minus any rollback transactions.

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 2 Number of Rows for Server

Table	Cardinality as built
warehouse	10,000
district	100,000
customer	300,000,000
history	300,000,000
new_order	300,000,000
orders	90,000,000
order_line	2,999,988,197
item	100,000
stock	1,000,000,000

No rows were deleted for the benchmark runs.

Database Layout

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

Table 3 depicts the database configuration of the tested and priced systems to meet the 8-hour steady state requirement.

Table 3 Data Distribution for benchmarked Configuration

Disk #	Drives	Partition	Size	Use
0		e: (NTFS)	53GB	OS
1	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs27 c:\tpcc\m27 u: (NTFS)	20GB 11GB 443GB	customer and stock miscellaneous mdf file
2	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs28 c:\tpcc\m28 n: (NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
3	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs29 c:\tpcc\m29 m:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file

4	14 - 36GB Adaptec SANbloc2 RAID1	c:\tpcc\cs30 c:\tpcc\m30 s:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
5	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs31 c:\tpcc\m31	20GB 11GB	customer and stock miscellaneous
6	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs32 c:\tpcc\m32 r:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
7	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs33 c:\tpcc\m33 w:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
8	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs34 c:\tpcc\m34 x:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
9	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs35 c:\tpcc\m35 y:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
10	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs36 c:\tpcc\m36 z:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
11	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs26 c:\tpcc\m26	20GB 11GB	customer and stock miscellaneous
12	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs13 c:\tpcc\m13	20GB 11GB	customer and stock miscellaneous
13	14 - 36GB Adaptec SANbloc2 RAID1	c:\tpcc\cs14 c:\tpcc\m14	20GB 11GB	customer and stock miscellaneous
14	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs15 c:\tpcc\m15	20GB 11GB	customer and stock miscellaneous
15	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs16 c:\tpcc\m16	20GB 11GB	customer and stock miscellaneous
16	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs17 c:\tpcc\m17	20GB 11GB	customer and stock miscellaneous
17	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs18 c:\tpcc\m18	20GB 11GB	customer and stock miscellaneous
18	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs19 c:\tpcc\m19	20GB 11GB	customer and stock miscellaneous
19	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs20 c:\tpcc\m20	20GB 11GB	customer and stock miscellaneous

20	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs21 c:\tpcc\m21	20GB 11GB	customer and stock miscellaneous
21	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs22 c:\tpcc\m22	20GB 11GB	customer and stock miscellaneous
22	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs23 c:\tpcc\m23	20GB 11GB	customer and stock miscellaneous
23	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs24 c:\tpcc\m24	20GB 11GB	customer and stock miscellaneous
24	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs25 c:\tpcc\m25	20GB 11GB	customer and stock miscellaneous
25	14 - 72GB Adaptec SANbloc2 RAID0+1	l:	474GB	Log File
26	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs1 c:\tpcc\m1	20GB 11GB	customer and stock miscellaneous
27	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs2 c:\tpcc\m2 h:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
28	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs3 c:\tpcc\m3	20GB 11GB	customer and stock miscellaneous
29	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs4 c:\tpcc\m4 i:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
30	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs5 c:\tpcc\m5	20GB 11GB	customer and stock miscellaneous
31	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs6 c:\tpcc\m6 j:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
32	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs7 c:\tpcc\m7	20GB 11GB	customer and stock miscellaneous
33	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs8 c:\tpcc\m8 k:(NTFS)	20GB 11GB 443GB	customer and stock miscellaneous backup file
34	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs9 c:\tpcc\m9	20GB 11GB	customer and stock miscellaneous
35	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs10 c:\tpcc\m10	20GB 11GB	customer and stock miscellaneous

36	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs11 c:\tpcc\m11	20GB 11GB	customer and stock miscellaneous
37	14 - 36GB Adaptec SANbloc2 RAID0	c:\tpcc\cs12 c:\tpcc\m12	20GB 11GB	customer and stock miscellaneous

Type of Database

A statement must be provided that describes:

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

Microsoft SQL Server 2005 Enterprise x64 Edition is a relational DBMS.

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

Database Mapping

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

The database was not replicated.

60 Day Space

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed.

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

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

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

The details of both the 8-hour transaction log space requirement and the 60-day space requirement are shown in Appendix D: 60-DAY SPACE.

Clause 5 Related Items

Throughput

Measured tpmC must be reported

Measured tpmC 125,954 tpmC
Price per tpmC 65.63 CNY/tpmC

Response Times

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

The TPC-C requirements for the average response time and the 90th percentile were met. **Table 4** provides the response times for each of the transaction types and the menu for the measured system.

Table 4 Response times in seconds

Response Times	90th Percentile	Average	Maximum
New-Order	0.22	0.17	5.09
Payment	0.17	0.13	2.44
Delivery	0.11	0.10	1.09
Stock Level	0.78	0.61	3.41
Order Status	0.19	0.16	5.04
Delivery (deferred)	0.13	0.10	5.00
Menu	0.11	0.10	1.51

Keying/Think Times

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

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

Table 5 Keying/think times

Keying/Think Times	Minimum	Average	Maximum
New-Order	18.02/0.00	18.04/12.07	18.55/121.01
Payment	3.02/0.00	3.03/12.06	3.55/121.01
Delivery	2.02/0.00	2.03/5.08	2.55/50.52
Stock Level	2.02/0.00	2.03/5.06	2.55/50.52
Order Status	2.02/0.00	2.03/10.05	2.54/100.53

Response Time Frequency Distribution Curves

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

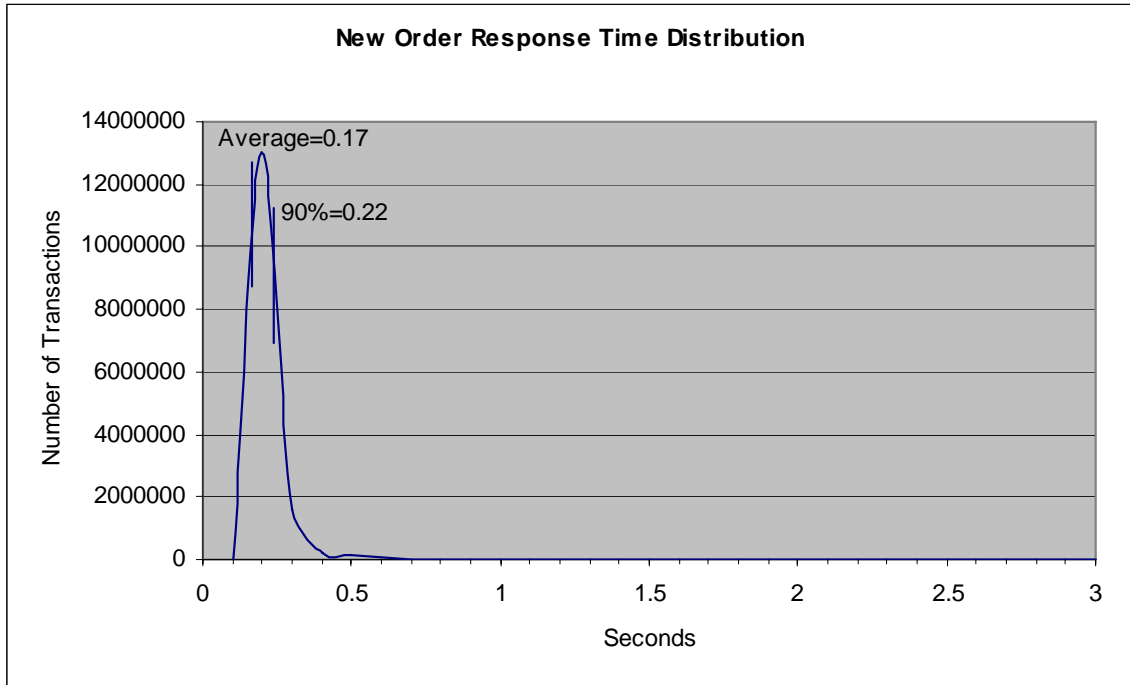


Figure 3 New order response time distribution

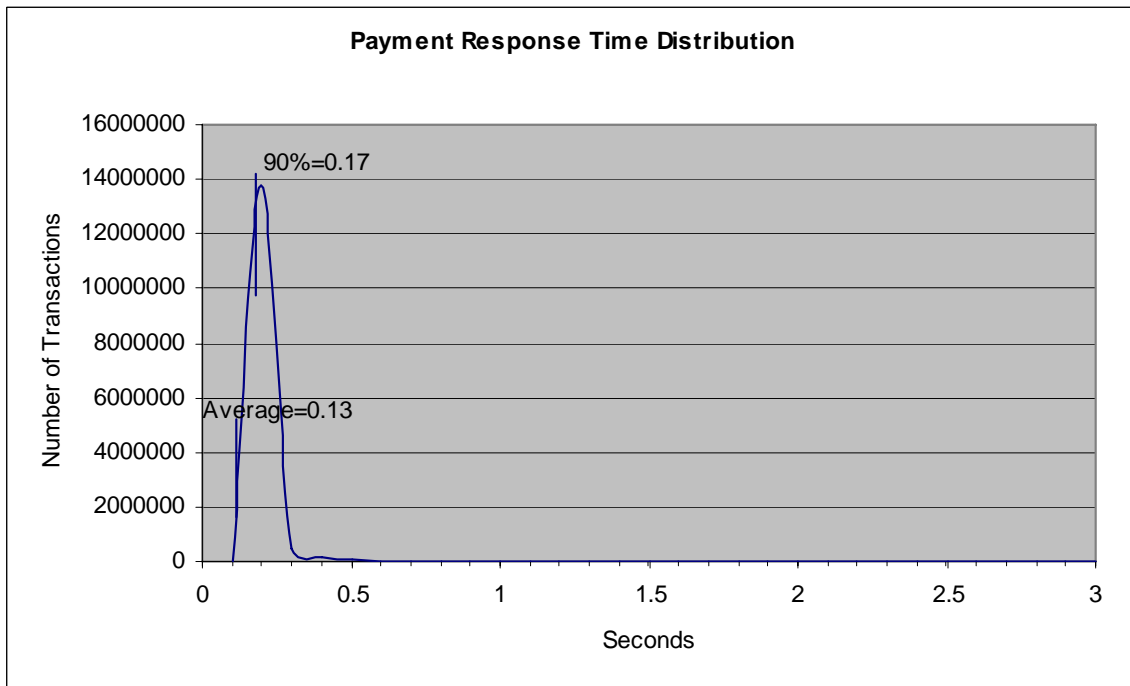


Figure 4 Payment response time Distribution

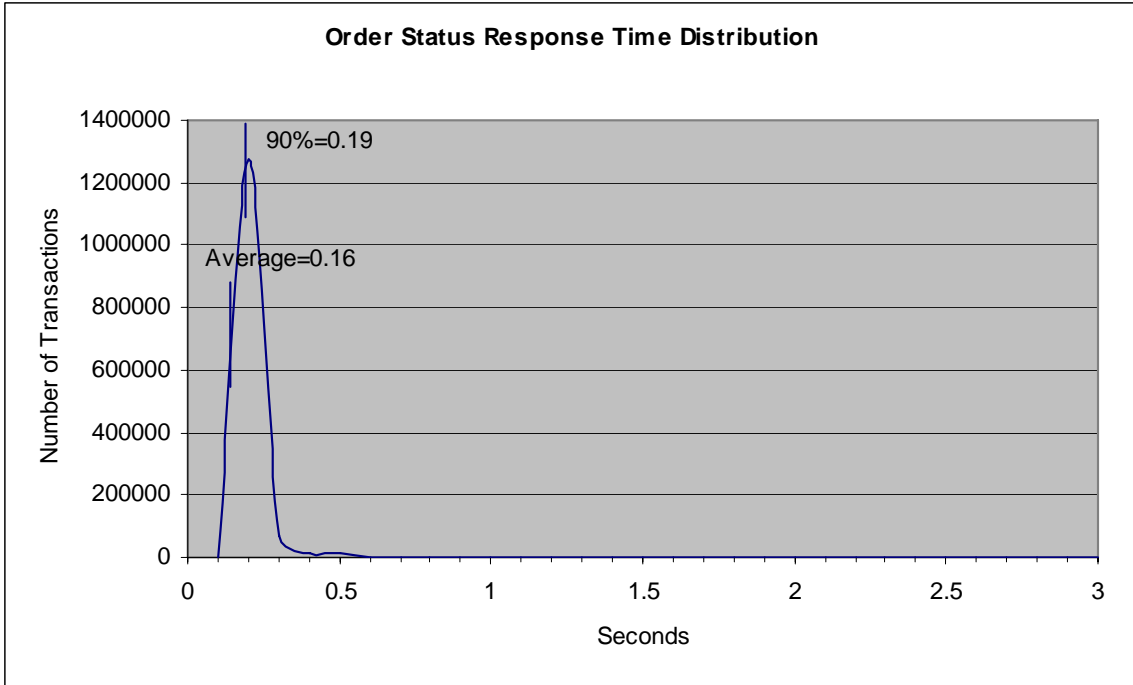


Figure 5 Order Status response time distribution

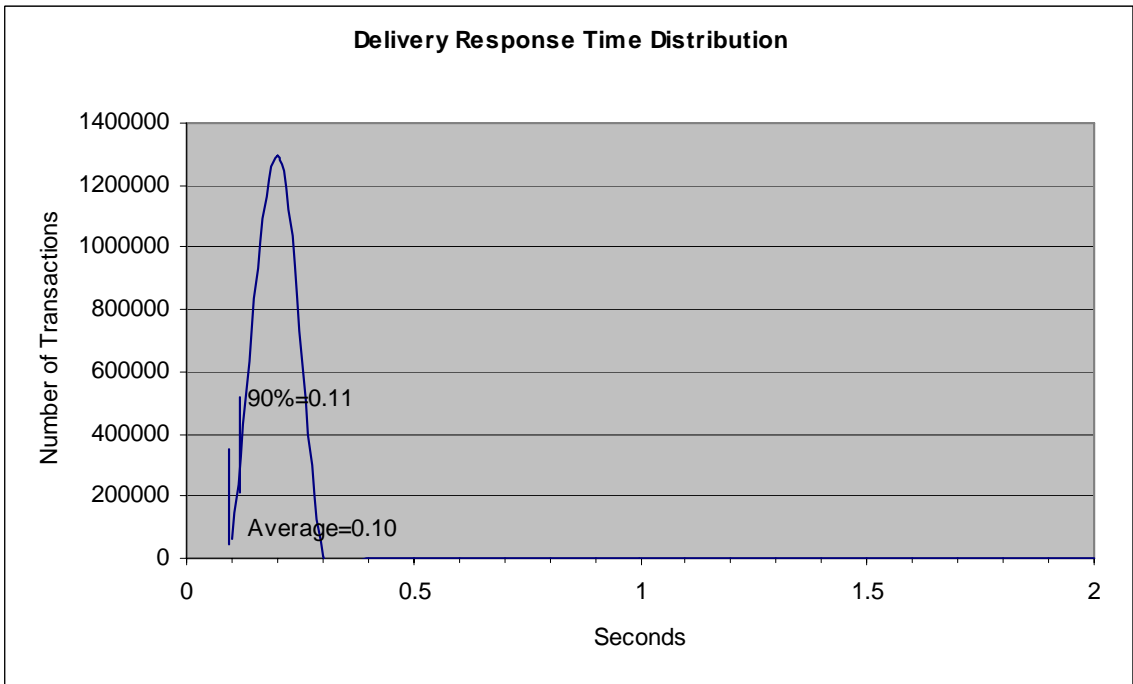


Figure 6 Delivery response time Distribution

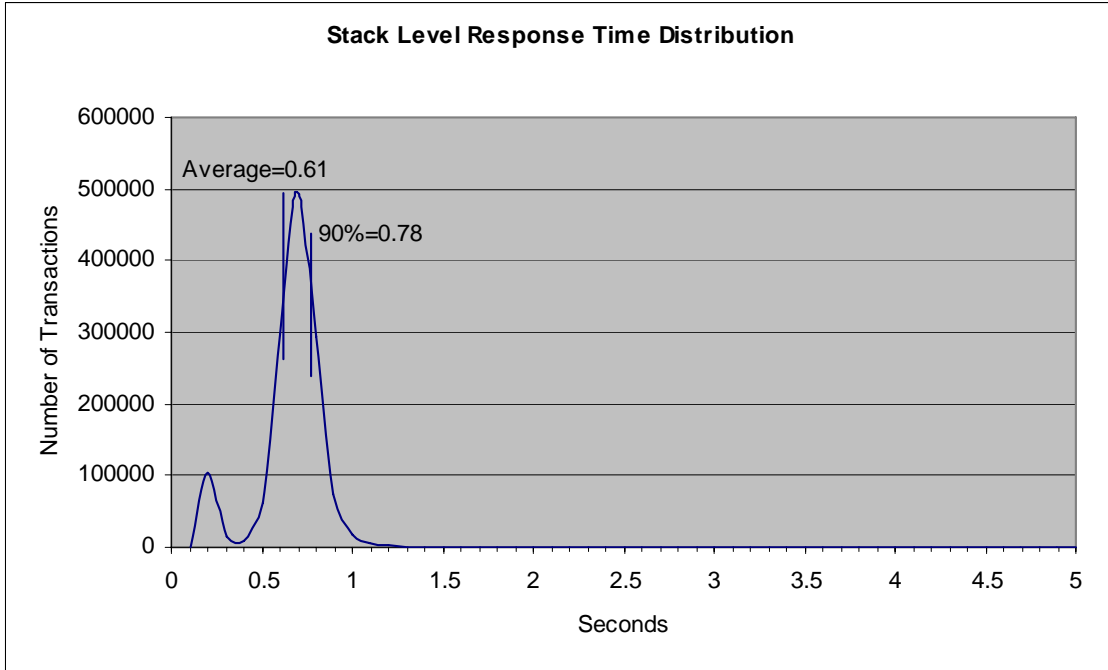


Figure 7 Stock level response time Distribution

Performance Curve for Response Time vs. Throughput

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

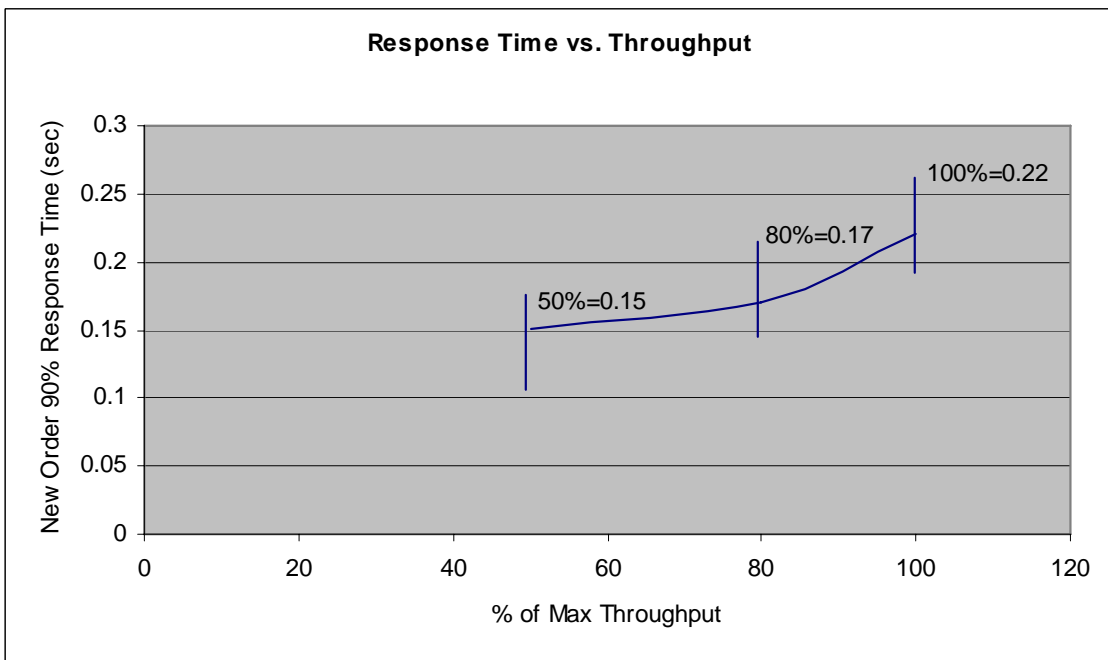


Figure 8 New Order response time vs. throughput

The frequency distribution of Think Times for the New-Order transaction n (see Clause 5.6.3), started and completed during the measurement interval, must be reported.

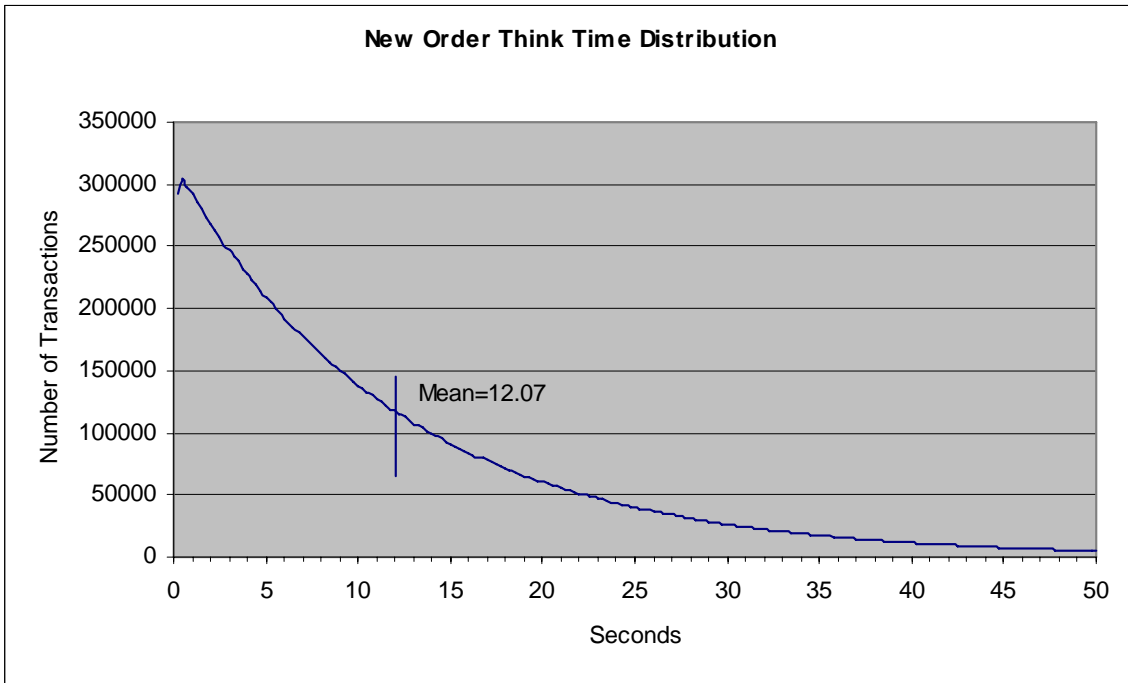


Figure 9 New Order think time Distribution

A graph (see Clause 5.6.4) of the throughput of the New-Order transaction versus elapsed time (i.e., wall clock) must be reported for both ramp-up time and measurement interval.

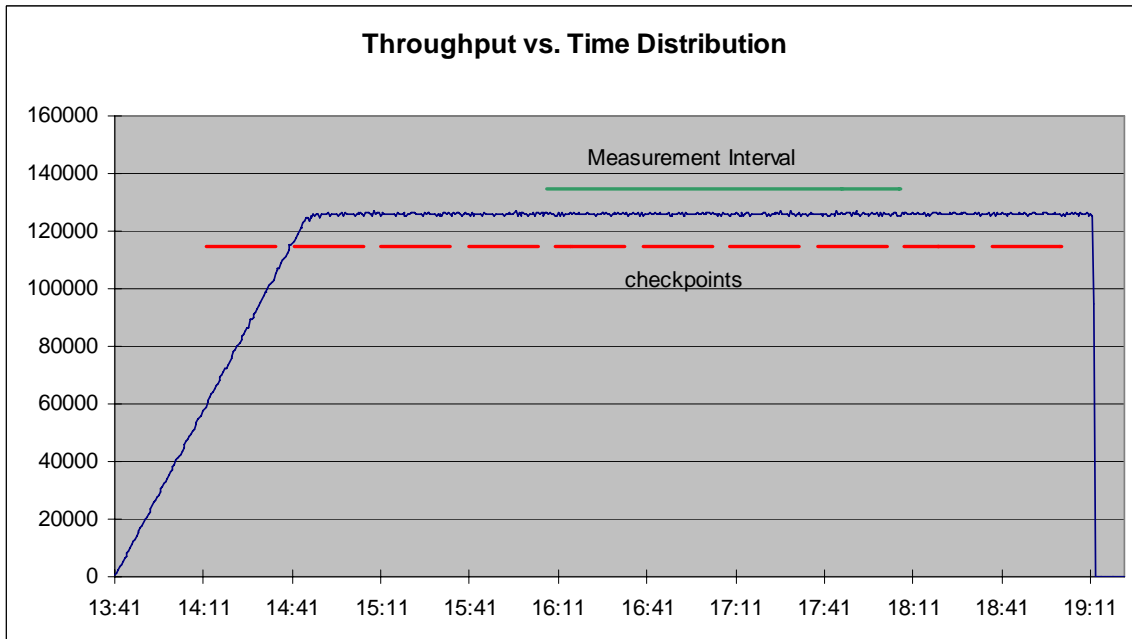


Figure 10 New Order Throughput vs. Elapsed Time

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

Figure 10 shows that the system was in steady state at the beginning of the measurement interval.

Work Performed During Steady State

A description of how the work normally performed during a sustained test (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 gigabit Ethernet LANs using DBLIB and RPC calls.

To perform checkpoints at specific intervals, the SQL Server recovery interval was set to 116 and a script was written to schedule multiple checkpoints at specific intervals. The script included a wait time between each checkpoint equal to 29 minutes 30 seconds so that the checkpoint interval was an integral multiple of the measurement interval, which was 120 minutes. The checkpoint script was started manually after the RTE had all users logged in and the database had achieved steady state.

At each checkpoint, Microsoft SQL Server wrote to disk all memory pages that had been updated but not yet physically written to disk. The positioning of the measurement interval is depicted on the graph in **Figure 10**.

Measurement Period Duration

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

The reported measured interval was exactly 120 minutes long.

Regulation of Transaction Mix

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

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

Transaction Statistics

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

See Table 6.

Table 6 Transaction Statistics and Transaction Mix

New Order	Value
Home warehouse order lines	99.00%
Remote warehouse order lines	1.00%
Rolled back transactions	1.00%
Average number of items per order	10.00
Payment	
Home warehouse payment transactions	84.99%
Remote warehouse payment transactions	15.01%
Non-Primary Key Access	
Payment transactions using C_LAST	60.01%
Order-Status transactions using C_LAST	60.02%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.74%
Payment	43.06%
Delivery	4.04%
Stock Level	4.08%
Order Status	4.08%

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 32 minutes after the start of the ramp-up. Subsequent checkpoints occurred every 29 minutes 30 seconds. The measurement interval contains four checkpoints. See **Table 7** for details.

Checkpoint Duration

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

The measure internal is from 16:07:00 to 18:07:00.

Table 7 Checkpoint Start time and Duration

Checkpoint Start Time	Duration (minutes)
16:09:52	24:00.0
16:39:22	24:00.0
17:08:53	24:00.0
17:38:22	24:00.0

Clause 6 Related Items

RTE Descriptions

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

The RTE used was Microsoft Benchcraft RTE. Benchcraft is a proprietary tool provided by Microsoft and is not commercially available. The RTE's input is listed in **Error! Reference source not found.**

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 4 SureServer R520. These driver machines emulated the users' web browsers.

Functional Diagrams

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

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

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

Networks

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

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

In the tested configuration, 4 driver (RTE) machines were connected through a gigabit Ethernet switch to the client machines at 1Gbs, thus providing the path from the RTEs to the clients. The server (SUT) was connected to the clients through the same gigabit Ethernet switch.

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

Operator Intervention

Systems must be able to run for at least 8 hours without operator intervention (see Clause 6.6.6).

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 the hardware and software used in the priced system must be reported. Each separately orderable item must have a vendor part number, description and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, vendor part number of the package and a description uniquely identifying each of the components of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported. The total 3-year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

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

Availability, Throughput, and Price Performance

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

- Maximum Qualified Throughput 125,954 tpmC
- Price per tpmC 65.63 CNY/tpmC
- Availability July 1, 2006

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 China PRC.

Usage Pricing

For any usage pricing, the sponsor must disclose:

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

The component pricing based on usage is shown below:

- 4 Microsoft Windows Server 2003, Standard Edition
- 1 Microsoft Windows Server 2003, Enterprise x64 Edition
- 2 Microsoft SQL Server 2005 Enterprise x64 Edition (per processor licensing)
- 1 Microsoft Visual C++ Standard Edition
- 3-year support for hardware components (except for components for which a minimum of 2 or 10 percent spares are provided)

Clause 8 Audit 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 TPC-C benchmark has been audited by Lorna Livingtree of Performance Metrics.

Availability of the Full Disclosure Report

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

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

Attestation Letter

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



PERFORMANCE METRICS INC.
TPC Certified Auditors

May 1st, 2006

Lei Feitao
Director, Technical Support & Service
Server Business, Lenovo
No.6 Shang Di Chuang Ye Road,
Hai Dian Distric,
Beijing, China

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

Platform: Lenovo T350
Database Manager: Microsoft SQL Server 2005 Enterprise x64 Edition
Operating System: Microsoft Windows 2003 Server, Enterprise x64 Edition
Transaction Monitor: Microsoft COM+

System Under Test:				
CPU's	Memory	Disks (total)	90% Response	TpmC
2 Intel Xeon @ 3.73 Ghz	Main: 32 GB	504 @ 36.4GB 14 @ 72GB 1 @ 120 GB (OS)	0.22	125,954
4 Clients each with:				
2 Intel Xeon @ 3.6 Ghz	Main: 2 GB	1 @ 76 GB	Na	Na

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

- The transactions were correctly implemented.
- The database files were properly sized and populated.
- The database was properly scaled with 10,000 warehouses, all of which were active during the measured interval.
- The ACID properties were successfully demonstrated.
- Input data was generated according to the specified percentages.

PERFORMANCE METRICS INC.
TPC Certified Auditors

- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 60 day space calculation was verified.
- The controller cache was disabled on the log disk controller.
- The steady state portion of the test was 120 minutes.
- At least one checkpoint was taken before the measured interval in steady state.
- Four complete checkpoints were taken during the measured interval.
- The system pricing was checked for major components and maintenance.

Auditor Notes:
None.

Sincerely,



Lorna Livingtree
Auditor

Appendix A: SOURCE CODE

Web Client Source Code

error.h

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

#pragma once

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

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

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

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

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

```

#define ERR_BAD_ITEM_ID 1
    //expected abort record in txnRecord
#define ERR_TYPE_DELIVERY_POST 2
    //expected delivery post failed
#define ERR_TYPE_WEBDLL 3
    //tpcc web generated error
#define ERR_TYPE_SQL 4
    //sql server generated error
#define ERR_TYPE_DBLIB 5
    //dblib generated error
#define ERR_TYPE_ODBC 6
    //odbc generated error
#define ERR_TYPE_SOCKET 7
    //error on communication socket client rte only
#define ERR_TYPE_DEADLOCK 8
    //dblib and odbc only deadlock condition
#define ERR_TYPE_COM 9
    //error from COM call
#define ERR_TYPE_TUXEDO 10
    //tuxedo error
#define ERR_TYPE_OS 11
    //operating system error
#define ERR_TYPE_MEMORY 12
    //memory allocation error
#define ERR_TYPE_TPCC_ODBC 13
    //error from tpcc odbc txn module
#define ERR_TYPE_TPCC_DBLIB 14
    //error from tpcc dblib txn module
#define ERR_TYPE_DELISRV 15
    //delivery server error
#define ERR_TYPE_TXNLOG 16
    //txn log error
#define ERR_TYPE_BCCONN 17
    //Benchcraft connection class
#define ERR_TYPE_TPCC_CONN 18
    //Benchcraft connection class
#define ERR_TYPE_ENCINA 19
    //Encina error
#define ERR_TYPE_COMPONENT 20
    //error from COM component
#define ERR_TYPE_RTE 21
    //Benchcraft rte
#define ERR_TYPE_AUTOMATION 22
    //Benchcraft automation errors
#define ERR_TYPE_DRIVER 23
    //Driver engine errors
#define ERR_TYPE_RTE_BASE 24
    //Framework errors
#define ERR_BUF_OVERFLOW 25
    //Buffer overflow during receive
#define ERR_TYPE_SOAP_HTTP 26
    //HTTP/SOAP dll generated error
#define ERR_TYPE_OLEDB 27

```

```

//OLE-DB generated error
#define ERR_TYPE_TPCC_OLEDB 28
//error from tpcc ole-db txn module
// TPC-W error types
#define ERR_TYPE_TPCW_CONN 50
//Benchcraft connection class
#define ERR_TYPE_TPCW_HTML 51
//error from TpcwHtml dll
#define ERR_TYPE_TPCW_USER 52
//error from TPC-W user class
#define ERR_TYPE_TPCW_ENG_BASE 53
#define ERR_TYPE_TPCW_ENG_OS 54
#define ERR_TYPE_HTML_RESP 55
#define ERR_TYPE_TPCW_ODBC 56
#define ERR_TYPE_SCHANNEL 57
#define ERR_TYPE_THINK_LIST 58
//----- end TPC-W -----
#define ERR_TYPE_XML_PROFILE 59
// TPC-E error types
#define ERR_TYPE_TPCE_CONN 60
//TPC-E pipe connection errors
#define ERR_TYPE_TPCE_RTE 61 //TPC-E
Rte errors
#define ERR_TYPE_TPCE_ENG_BASE 62 //Tpce
Driver engine errors
#define ERR_TYPE_TPCE_ENG_OS 63 //Tpce
Driver engine system errors
//#define ERR_TYPE_TPCE_MEE_ENG_BASE 64 //Tpce
MEE Driver engine errors
//#define ERR_TYPE_TPCE_MEE_ENG_OS 65 //Tpce
MEE Driver engine system errors

#define ERR_INS_MEMORY "Insufficient Memory to continue."
#define ERR_UNKNOWN "Unknown error."
#define ERR_MSG_BUF_SIZE 512
#define INV_ERROR_CODE -1
#define ERR_INS_BUF_OVERFLOW "Insufficient Buffer size to receive HTML pages."

```

```

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

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

```

```

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

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

```

```

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

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

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

```

```

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

```

```

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

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

```

```

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

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

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

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

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

    //short  m_errType;
};

class CSocketErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eSend,
        eSocket,
        eBind,
        eConnect,
        eListen,
        eHost,
        eRecv,
        eGetHostByName,
        eWSACreateEvent,
        eWSASend,
        eWSAGetOverlappedResult,
        eWSARecv,
        eWSAWaitForMultipleEvents,
        eWSAStartup,
        eWSAResetEvent,
        eWSAEnumNetworkEvents,
        eWSAEventSelect,
        eSelect,
        eAccept,
        eNonRetryable
    };
};

```



```

CSocketErr(Action eAction, LPCTSTR szLocation = NULL);

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

Action m_eAction;
char *m_szErrorText;

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

};

class CSystemErr : public CBaseErr
{
public:
    enum Action
    {
        eNone = 0,
        eTransactNamedPipe,
        eWaitNamedPipe,
        eSetNamedPipeHandleState,
        eCreateFile,
        eCreateProcess,
        eCallNamedPipe,
        eCreateEvent,
        eCreateThread,
        eVirtualAlloc,
        eReadFile = 10,
        eWriteFile,
        eMapViewOfFile,
        eCreateFileMapping,
        eInitializeSecurityDescriptor,
        eSetSecurityDescriptorDacl,
        eCreateNamedPipe,
        eConnectNamedPipe,
        eWaitForSingleObject,
        eRegOpenKeyEx,
        eRegQueryValueEx = 20,
        ebeginthread,
        eRegEnumValue,
        eRegSetValueEx,
        eRegCreateKeyEx,
        eWaitForMultipleObjects,
        eRegisterClassEx,
        eCreateWindow,
        eCreateSemaphore,
    };
};

```

```

        eReleaseSemaphore,
        eFSeek,
        eFRead,
        eFWrite,
        eTmpFile,
        eSetFilePointer,
        eNew,
        eCloseHandle,
        eGetOverlappedResult
    };

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

    Action      m_eAction;

private:
    char m_szMsg[ERR_MSG_BUF_SIZE];
};

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

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

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

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

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

```

```

        LoadSchema,
        ValidateProfile,
        SaveProfile,
        LoadFromXML,
        SaveToXML,
        ApplyProcessingInstruction,
        ApplyAttribute,
        ApplyNode
    };

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

    virtual int         ErrorType() { return ERR_TYPE_XML_PROFILE; };
    virtual char       *ErrorTypeStr() { return "XML PROFILE"; };
    virtual char       *ErrorText();
    virtual int         ErrorCode() { return m_eCode; };
    int                 ErrorAction() { return (int)m_eAction; }
    //virtual void      Draw(HWND hwnd, LPCTSTR szStr = NULL)
    //{
    //    ::MessageBox(hwnd, szStr, m_szLoc, MB_OK);
    //};
private:
    char    m_szMsg[ERR_MSG_BUF_SIZE];
    LPCTSTR m_szLoc;
    int     m_eCode;
    bool    m_bOverload;
    Action m_eAction;

};

```

ReadRegistry.cpp

```

/*      FILE:          READREGISTRY.CPP
 *
 *          Microsoft TPC-C Kit Ver. 4.20.000
 *          Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *
 *          not yet audited
 *
 */

```

```

*      PURPOSE:      Implementation for TPC-C Tuxedo class.
*      Contact: Charles Levine (clevine@microsoft.com)
*
* Change history:
*      4.20.000 - first version
*/

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

```

```

&size) != ERROR_SUCCESS )
    pReg->szDbPassword[0] = 0;

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

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

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

    RegCloseKey(hKey);

    return FALSE;
}

```

ReadRegistry.h

```

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

```

```

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

```

```

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

```

//This structure defines the data necessary to keep distinct for each terminal or client connection.

```

typedef struct _TPCCREGISTRYDATA
{
    enum DBPROTOCOL eDB_Protocol;

```

```

enum TXNMON eTxnMon;
BOOL bCOM_SinglePool;
DWORD dwMaxConnections;
DWORD dwMaxPendingDeliveries;
DWORD dwNumberOfDeliveryThreads;
char szPath[128];
char szDbServer[32];
char szDbName[32];
char szDbUser[32];
char szDbPassword[32];
wchar_t szSPPrefix[32]; //tpcc_odbc.dll stored procedures prefix
DWORD dwConnectDelay; // delay in ms to use in pacing connection open and close
BOOL bCallNoDuplicatesNewOrder; // whether to check for non-duplicate item ids and
call a different New Order SP
} TPCCREGISTRYDATA, *PTPCCREGISTRYDATA;

BOOL ReadTPCCRegistrySettings( TPCCREGISTRYDATA *pReg );

```

trans.h

```

/* FILE: TRANS.H
 * Microsoft TPC-C Kit Ver. 4.42.000
 * Copyright Microsoft, 2002
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 * PURPOSE: Header file for TPC-C structure templates.
 *
 * Change history:
 * 4.42.000 - changed w_id fields from short to long to support >32K warehouses
 * 4.20.000 - updated rev number to match kit
 */
#pragma once

// String length constants
#define SERVER_NAME_LEN 20
#define DATABASE_NAME_LEN 20
#define USER_NAME_LEN 20
#define PASSWORD_LEN 20
#define TABLE_NAME_LEN 20
#define I_DATA_LEN 50
#define I_NAME_LEN 24
#define BRAND_LEN 1
#define LAST_NAME_LEN 16
#define W_NAME_LEN 10
#define ADDRESS_LEN 20
#define STATE_LEN 2
#define ZIP_LEN 9
#define S_DIST_LEN 24
#define S_DATA_LEN 50

```

```

#define D_NAME_LEN          10
#define FIRST_NAME_LEN     16
#define MIDDLE_NAME_LEN    2
#define PHONE_LEN          16
#define DATETIME_LEN       30
#define CREDIT_LEN         2
#define C_DATA_LEN         250
#define H_DATA_LEN         24
#define DIST_INFO_LEN      24
#define MAX_OL_NEW_ORDER_ITEMS  15
#define MAX_OL_ORDER_STATUS_ITEMS 15
#define STATUS_LEN         25
#define OL_DIST_INFO_LEN    24

// TIMESTAMP_STRUCT is provided by the ODBC header file sqltypes.h, but is not available
// when compiling with dblib, so redefined here. Note: we are using the symbol "__SQLTYPES"
// (declared in sqltypes.h) as a way to determine if TIMESTAMP_STRUCT has been declared.
#ifndef __SQLTYPES
    typedef struct
    {
        short          /* SQLSMALLINT */   year;
        unsigned short /* SQLUSMALLINT */  month;
        unsigned short /* SQLUSMALLINT */  day;
        unsigned short /* SQLUSMALLINT */  hour;
        unsigned short /* SQLUSMALLINT */  minute;
        unsigned short /* SQLUSMALLINT */  second;
        unsigned long  /* SQLINTEGER */    fraction;
    } TIMESTAMP_STRUCT;
#endif

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

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

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

```



```

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

    // output params
    EXEC_STATUS exec_status_code;
    char        c_last[LAST_NAME_LEN+1];
    char        c_credit[CREDIT_LEN+1];
    double      c_discount;
    double      w_tax;
    double      d_tax;
    long        o_id;
    short       o_commit_flag;
    TIMESTAMP_STRUCT o_entry_d;
    short       o_all_local;
    double      total_amount;
    OL_NEW_ORDER_DATA OL[MAX_OL_NEW_ORDER_ITEMS];
} NEW_ORDER_DATA, *PNEW_ORDER_DATA;

```

```

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

    // output params
    EXEC_STATUS exec_status_code;
    TIMESTAMP_STRUCT h_date;
    char        w_street_1[ADDRESS_LEN+1];
    char        w_street_2[ADDRESS_LEN+1];
    char        w_city[ADDRESS_LEN+1];
    char        w_state[STATE_LEN+1];
    char        w_zip[ZIP_LEN+1];
    char        d_street_1[ADDRESS_LEN+1];
    char        d_street_2[ADDRESS_LEN+1];
    char        d_city[ADDRESS_LEN+1];
    char        d_state[STATE_LEN+1];
    char        d_zip[ZIP_LEN+1];
    char        c_first[FIRST_NAME_LEN+1];
    char        c_middle[MIDDLE_NAME_LEN + 1];
    char        c_street_1[ADDRESS_LEN+1];
    char        c_street_2[ADDRESS_LEN+1];
    char        c_city[ADDRESS_LEN+1];
    char        c_state[STATE_LEN+1];

```

```

char                c_zip[ZIP_LEN+1];
char                c_phone[PHONE_LEN+1];
TIMESTAMP_STRUCT  c_since;
char                c_credit[CREDIT_LEN+1];
double             c_credit_lim;
double             c_discount;
double             c_balance;
char                c_data[200+1];
} PAYMENT_DATA, *PPAYMENT_DATA;

```

```

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

```

```

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

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

```

```

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

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

```

//This structure is used for posting delivery transactions and for writing them to the delivery server.

```
typedef struct _DELIVERY_TRANSACTION
```

```

{
    SYSTEMTIME queue;           //time delivery transaction queued
    long        w_id;           //delivery warehouse
    short       o_carrier_id;   //carrier id
} DELIVERY_TRANSACTION;

```

```

typedef struct
{
    // input params
    long        w_id;
    short       d_id;
    short       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;
};

```

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

txnlog.h
/*      FILE:          TXNLOG.H
 *
 *                      Microsoft TPC-C Kit Ver. 4.10.000
 *                      not yet audited
 *
 *      PURPOSE:      Header file for txn log class
 *                      Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 */
#include <stdio.h>          //needed for FILE

#define DRIVER_NAME_LEN          32          //max length of the driver
engine name - must be the same as in engstut.h!
#define TXN_LOG_INCORRECTLY_SHUT_DOWN    100    //ctrl rec subtype generated by the
txn log when reading an abruptly shut down log

#pragma once

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

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

typedef struct _TXN_ORDERSTATUS
{
    BYTE    CustByName;
} TXN_ORDERSTATUS;

```

```

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

// Common header for all records in txn log. The TxnType field is
// a switch which identifies the particular variant.
#define TXN_REC_TYPE_CONTROL    1    //
#define TXN_REC_TYPE_TPCC      2    // replaces
TRANSACTION_TYPE_TPCC
#define TXN_REC_TYPE_TPCC_DELIV_DEF 3
#define TXN_REC_TYPE_TPCW      4    // replaces
TRANSACTION_TYPE_TPCW

typedef struct _TXN_RECORD_HEADER
{
    JULIAN_TIME TxnStartT0;    // start of txn
    BYTE TxnType;             // one of TXN_REC_TYPE_*
    BYTE TxnSubType;          // depends on TxnType
} TXN_RECORD_HEADER, *PTXN_RECORD_HEADER;

typedef struct _TXN_RECORD_CONTROL
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME TxnStartT0;    // start of txn
    BYTE TxnType;             // = TXN_REC_TYPE_CONTROL
    BYTE TxnSubType;          // depends on TxnType
    // end of common header

    DWORD Len;                // number of bytes after this field
} TXN_RECORD_CONTROL, *PTXN_RECORD_CONTROL;

// TPC-C Txn Record Layout:
//
//"TxnStartT0" is a Julian timestamp corresponding to the moment the
//txn is sent to the SUT, i.e., beginning of response time. Deltas
//are in milliseconds. Note that if RTDelay > 0, then the txn was
//delayed by this amount. The delay occurs at the beginning of the
//response time. So if RTDelay > 0, then the txn was actually sent
//at TxnStartT0 + RTDelay.
//
//Graphically:
//
// time -->
//
// |--- Menu ---|-- Keying --|-- Response --|--- Think --|
// <- DeltaT1 -> <- DeltaT2 -> <- DeltaT4 -> <- DeltaT3 ->
//
//

```



```

//          ^ TxnStartT0
//
//RTDelay is the amount of response time delay included in DeltaT4.
//RTDelay is recorded per txn because this value can be changed on
//the fly, and so may vary from txn to txn.
//
//TxnStatus is the txn completion code. It is used to indicate errors.
//For example, in the New Order txn, 1% of txns abort. TxnStatus will
//reflect this.

typedef struct _TXN_RECORD_TPCC
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME TxnStartT0;          // start of txn
    BYTE TxnType;                    // = TXN_REC_TYPE_TPCC
    BYTE TxnSubType;                // depends on TxnType
    // end of common header

    int DeltaT1;                    // menu time (ms)
    int DeltaT2;                    // keying time (ms)
    int DeltaT3;                    // think time (ms)
    int DeltaT4;                    // response time (ms)
    int RTDelay;                    // response time delay (ms)
    int TxnError;                   // error code providing more detail for
TxnStatus
    int w_id;                        // warehouse ID
    BYTE d_id;                      // assigned district ID for this thread
    BYTE d_id_ThisTxn;              // district ID chosen for this particular
    BYTE TxnStatus;                 // completion status for txn to indicate errors
    BYTE reserved;                  // for word alignment
    TXN_DETAILS TxnDetails;         //

    bool IsSuccessRecord() { return (TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus == ERR_TYPE_DELIVERY_POST); }
} TXN_RECORD_TPCC, *PTXN_RECORD_TPCC;

// TPC-C Deferred Delivery Txn Record Layout:
//
//Incorporating delivery transaction information into the above
//structure would increase the size of TXN_DETAILS from 8 to 42 bytes.
//Hence, we store delivery transaction details in a separate structure.
//
typedef struct _TXN_RECORD_TPCC_DELIV_DEF
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME TxnStartT0;          // start of txn
    BYTE TxnType;                    //
TXN_REC_TYPE_TPCC_DELIV_DEF
    BYTE TxnSubType;                // = 0
    // end of common header

    int DeltaT4;                    // response time (ms)
    int DeltaTxnExec;               // execution time (ms)

```

```

int          w_id;                // warehouse ID
BYTE TxnStatus;                // completion status for txn to indicate errors
BYTE reserved;                // for word alignment
short o_carrier_id;           // carrier id
long  o_id[10];                // returned delivery transaction ids

bool IsSuccessRecord() { return (TxnStatus == ERR_SUCCESS || TxnStatus ==
ERR_BAD_ITEM_ID || TxnStatus == ERR_TYPE_DELIVERY_POST); }
} TXN_RECORD_TPCC_DELIV_DEF, *PTXN_RECORD_TPCC_DELIV_DEF;

//
//TPC-W records.
//
typedef struct _TXN_RECORD_TPCW
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME TxnStartT0;        // start of txn
    BYTE TxnType;                 // = TXN_REC_TYPE_TPCW
    BYTE TxnSubType;              // depends on TxnType
    // end of common header

    int ThinkTime;                // think time (ms)
    int WIRT;                      // response time (ms)
    int TxnError;                 // error code providing more detail for
TxnStatus
    BYTE TxnStatus;                // completion status for txn to indicate errors
    //This field below depends on the txn sub type:
    // - for Home interaction: it indicates whether the user was a new customer (or
returning)
    // - for Buy Confirm:          it indicates whether the shipping address was
updated
    // - for Search Request:       it indicates the search type (Author, Title, or
Subject)

    //This statistics needs to be reported according to 5.5.5.1 clause in the specs.
    //Because this field occupies 1 byte, the record structure is already aligned on word
boundary.
    union {
        BYTE newCustomer;
        BYTE addrUpdated;
        BYTE searchType;
    } intrDetails;

    //This field is mostly for informational/debugging purposes.
    //It indicates what user performed this web interaction and what instance (session) of
that use it was.
    //The first 22 bits indicate the user #, and the top 10 bits indicate instance (session) #.
    unsigned __int32 uiUser;

    bool IsSuccessRecord() { return (TxnStatus == ERR_SUCCESS); }
} TXN_RECORD_TPCW, *PTXN_RECORD_TPCW;

//
// Data part of a control record written when a user is created (or it's new session) - to

```

```

record USMD
    typedef struct _TXN_RECORD_TPCW_USER_DATA
    {
        unsigned __int32    uiUser;                // user number
        JULIAN_TIME        USMD;                // USMD for this
user
        BYTE                bRetCust;            // returning
customer?
    } TXN_RECORD_TPCW_USER_DATA, *PTXN_RECORD_TPCW_USER_DATA;

//The entire TPCW User control record structure
typedef struct _TXN_RECORD_TPCW_USER
{
    // common header; must exactly match TXN_RECORD_HEADER
    JULIAN_TIME TxnStartT0;                // start of txn
    BYTE TxnType;                // = TXN_REC_TYPE_CONTROL
    BYTE TxnSubType;                // depends on TxnType
    // end of common header

    DWORD Len;                // number of bytes after this field
    //The fields above must exactly match TXN_RECORD_CONTROL

    //The fields below must exactly match TXN_RECORD_TPCW_USER_DATA
    unsigned __int32    uiUser;                // user number
    JULIAN_TIME        USMD;                // USMD for this
user
    BYTE                bRetCust;            // returning
customer?
    } TXN_RECORD_TPCW_USER, *PTXN_RECORD_TPCW_USER;

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

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

////////////////////////////////////
// The transaction log has a header as the first 4K block.
//
typedef struct _TXN_LOG_HEADER
{
    char EyeCatcher[2]; // signature bytes; should always be "BC"
    int LogVersion; // set to
TXN_LOG_VERSION
    JULIAN_TIME BeginTxnTS; // timestamp of first (lowest) txn

```

```

start
    JULIAN_TIME      EndTxnTS;          // timestamp of last (highest) txn
completion time
    int              iRecCount;         // number of records in
log file
    BOOL             bLogSorted;
    int              iFileSize;         // file size in bytes

    // driver engine that created this log file
    char             szDriverEngineName[DRIVER_NAME_LEN];
    // the record map provides a fast way to get close to a particular timestamp in a sorted
log file.
//
// struct
//
//     JULIAN_TIME      TS;              // timestamp of record
//     int              iPos;            // byte position in file
// }
// RecMap[RecMapSize];
// #define             RecMapSize       200

} TXN_LOG_HEADER, *PTXN_LOG_HEADER;

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

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

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

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

#define TXN_LOG_OS_ERROR1
#define TXN_LOG_NOT_SORTED   2

#define SKIP_CTRL_RECS       1

class CTxnLog

```

```

{
    private:
        DWORD          iBufferSize;          //buffer allocated size
        DWORD          iBytesFreeInBuffer;   //total bytes available for
use in buffer
        int            iNumBuffers;          //buffers in use
        int            iActiveBuffer;        //indicates which buffer
is active: 0 or 1
        int            iIoBuffer;           //buffer for any
pending IO operation
//
        int            iFilePointer;         //position in file.
        LARGE_INTEGER  lFilePointer;         //position in file.
        int            iNextRec;            //when reading,
ordinal value of next record

        // A "save point" is remembered each time GetNextRecord is called with a start time
specified.
        // The next time it is called, if start time is after the save point, we start scanning from
the
        // save point. This is particularly useful in FindBestInterval, where the log is scanned
repeatedly.
        JULIAN_TIME    SavePtTime;
//
        int            iSavePtFilePointer;
        LARGE_INTEGER  lSavePtFilePointer;
        int            iSavePtNextRec;

        JULIAN_TIME    lastTS;               //when writing sorted
output, used to verify records are sorted
        BOOL           bWrite;               //writing log file
        BOOL           bCrashOpen;          // tolerate bad headers
and consistency checks

        BOOL           bLogSorted;          // is log file sorted?
applies to both input and output
        JULIAN_TIME    BeginTxnTS;          // timestamp of first
(lowest) txn start
        JULIAN_TIME    EndTxnTS;           // timestamp of last
(highest) txn completion time
        int            iRecCount;          // number of
records in log file
        // To write a checkpoint information into the header, need to know the EndTxnTS for
the
        // last record written to the disk. It is not necessarily the last record in the
// last written buffer, as the last record may be only partially in the buffer.
        // So remember the timestamps for 2 last records that begin in the buffer - one of
// them will be the last complete record written to disk.
        JULIAN_TIME    PrevEndTxnTS;       // timestamp of the previous to
last record
        union {
            TXN_LOG_HEADER  HeaderForCheckpoint; // header written on
every checkpoint
            char             szHeaderBuffer[512]; // 512 bytes is
the minimum we can write to the disk

```

```

    } HeaderBuffer; //need the union because can't write sizeof(TXN_LOG_HEADER) -
too few bytes

// Control record returned from GetNextRecord if the file
// currently opened for read was not properly shut down
struct
{
    TXN_RECORD_CONTROL    RecHeader;
    char                  szDriverName[DRIVER_NAME_LEN];

} IncorrectShutDownRec;

BYTE    *pCurrent;           //ptr to current buffer
BYTE    *pBuffer[MAX_NUM_BUFFERS];

PTXN_RECORD_HEADER    *TxnArray;           //transaction record
pointer array for sort

DWORD    dwError;
DWORD    dwCheckpointError;           //error in checkpoint
thread

HANDLE    hTxnFile;           //handle to log
file

HANDLE    hMapFile;           //map file used
when sorting the log

HANDLE    hIoComplete;           //event to signify that
there are no pending IOs

HANDLE    hLogFileIo;           //event to signal
the IO thread to write the inactive buffer

HANDLE    hStopCheckpointThread; //event to signal the checkpoint
thread to exit

Spinlock    Spin;           //spin lock to protect the
txn log file buffers

Spinlock    WriteSpin;           //spin lock to protect the
WriteFile operation between IO and Checkpoint threads

FILE    *tmpFile;           //temp file for merging
sorted pieces

PBLOCK_HEADER    tmpHeaders;           //sorted pointers block
header

BYTE    **recPointers; //record pointer buffers for each
sorted block

PTXN_RECORD_HEADER    *recBuffers; //record buffers for each sorted
block

int    *PointersRead;           //# of pointers processed
in each block

BOOL    *BlockAvailable;           //whether to check a particular
block for jmin

int    nBlocks;
int    jmin;           //index (block-
wise) of the lowest timestamp record

```

```

        int          iAvgRecordLen;                //average record length

        int          iSortedReturnedCount;        //keeps track of the # of sorted
records returned through GetSortedRecord()

        BOOL  bIncorrectShutDown;                // indicates whether the log
opened for read was not correctly shut down

        int Write(BYTE *ptr, DWORD Size);
        static void LogFileIO(CTxnLog *);

        void LoadBuffers(int j);                //used in sort/merge to load record buffers

        static void CheckpointThread(CTxnLog *);    // checkpointing thread

public:

        CTxnLog(LPCTSTR szFileName, DWORD dwOpts, char *szDriver = NULL);
        ~CTxnLog(void);

        int WriteToLog(PTXN_RECORD_TPCC pTxnRcrd);
        int WriteToLog(PTXN_RECORD_TPCC_DELIV_DEF pTxnRcrd);
        int WriteToLog(PTXN_RECORD_CONTROL pCtrlRec);
        int WriteToLog(PTXN_RECORD_HEADER pCtrlRec);
        int WriteToLog(PTXN_RECORD_TPCW pTxnRcrd);    //support for TPC-W

        int WriteCtrlRecToLog(BYTE SubType, LPTSTR lpStr, DWORD dwLen);

        void CloseTransactionLogFile(void);

        PTXN_RECORD_HEADER GetNextRecord(BOOL bSkipCtrlRecs = FALSE);
        PTXN_RECORD_HEADER GetNextRecord(JULIAN_TIME SeekTimeT0, BOOL
bSkipCtrlRecs = FALSE);

        int Sort(void);
        PTXN_RECORD_HEADER GetSortedRecord();

        inline BOOL IsSorted(void) { return bLogSorted; };
        inline JULIAN_TIME BeginTS(void) { return BeginTxnTS; };
        inline JULIAN_TIME EndTS(void) { return EndTxnTS; };
        inline int RecordCount(void) { return iRecCount; };
};

class CTXNLOG_ERR : public CBaseErr
{
public:
    enum CTXNLOG_ERRS
    {
        ERR_BAD_FILE_FORMAT,                // "File format is invalid."
        ERR_UNKNOWN_LOG_VERSION,            // "Log file version is unknown."
        ERR_BROKEN_LOG_FILE,                // "Log file is broken."
        ERR_LOG_NOT_SORTED,                  // "Log file is not sorted"
        ERR_INVALID_TIME_SEQ,                // "Internal Error: Record Time

```

```

Sequence invalid."
    };

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

    int ErrorType() {return ERR_TYPE_TXNLOG;};
    char *ErrorTypeStr() { return "TXN LOG"; }

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

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

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

```

tpcc_dblib.cpp

```

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

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

```



```

#include <assert.h>

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

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

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

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

#define DEFCLPACKSIZE                4096

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

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

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

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

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

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

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

```

```

        CTPCC_DBLIB          *pConn;

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

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

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

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

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

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

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

    return 0;
}

```

```

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

```

```

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

    return;
}

```

```

/* FUNCTION: CTPCC_DBLIB_ERR::ErrorText
*
*/

```

```

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

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

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

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_errno == errorMsgs[i].iError )

```

```

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

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

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

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

    m_MaxRetries = 10;          // how many retries on deadlock

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

    // allocate a login structure
    login = dblogin();
    if (login == NULL)
        ThrowError(CDBLIBERR::eLogin);
    InterlockedIncrement( &iConnectionCount );

    // register error and message handler functions

```

```

if (dbprocerrhandle(login, err_handler) == NULL)
    ThrowError(CDBLIBERR::eDbProcHandler);

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

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

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

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

m_dbproc = dbopen(login, szServer);

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

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

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

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

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

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

DiscardNextResults(2);

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

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

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

```

```

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

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

        DiscardNextRows(0);
        DiscardNextResults(0);
    }

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

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

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

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

void CTPCC_DBLIB::SetSqlError( int /*DBINT*/ msgno, int msgstate, int severity, LPCSTR
msgtext )

```

```

{
    if (m_SqlErr == NULL)
        m_SqlErr = new CSQLERR();

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

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

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

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

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

    throw pDbLibErr;
}

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

```

```

int          iRowsRead = 0;
RETCODE     rc;

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

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

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

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

        DiscardNextRows(-1);
        iResultsRead++;
    }

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

```



```

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

    ResetError();

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

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

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

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

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

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

            DiscardNextRows(0);
            DiscardNextResults(0);

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

```

```

    } // while (TRUE)

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

```

```

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

    int iTryCount = 0;
    const BYTE *pData;

    ResetError();

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

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

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

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

```

```

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

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

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

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

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

        if(pData=dbdata(m_dbproc, 1))
            UtilStrCpy(m_txn.NewOrder.OL[i].ol_i_name,      pData,
dbdatlen(m_dbproc, 1));

        if(pData=dbdata(m_dbproc, 2))
            m_txn.NewOrder.OL[i].ol_stock = (*(DBSMALLINT *)
pData);

        if(pData=dbdata(m_dbproc, 3))
            UtilStrCpy(m_txn.NewOrder.OL[i].ol_brand_generic,
pData, dbdatlen(m_dbproc, 3));

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

        if(pData=dbdata(m_dbproc, 5))

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

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

        DiscardNextRows(0);
    }

    // get remaining values for w_tax, d_tax, o_id, c_last, c_discount, c_credit,

```

```

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

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

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

if (pData=dbdata(m_dbproc, 1))
    dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,1), SQLFLT8, (BYTE *)&m_txn.NewOrder.w_tax, 8);
if (pData=dbdata(m_dbproc, 2))
    dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,2), SQLFLT8, (BYTE *)&m_txn.NewOrder.d_tax, 8);
if (pData=dbdata(m_dbproc, 3))
    m_txn.NewOrder.o_id = *(DBINT *) pData);
if (pData=dbdata(m_dbproc, 4))
    UtilStrCpy(m_txn.NewOrder.c_last, pData, dbdatlen(m_dbproc, 4));
if (pData=dbdata(m_dbproc, 5))
    dbconvert(m_dbproc, SQLNUMERIC, (LPCBYTE)pData,
dbdatlen(m_dbproc,5), SQLFLT8, (BYTE *)&m_txn.NewOrder.c_discount, 8);
if (pData=dbdata(m_dbproc, 6))
    UtilStrCpy(m_txn.NewOrder.c_credit, pData, dbdatlen(m_dbproc,
6));
if (pData=dbdata(m_dbproc, 7))
{
    datetime = *((DBDATETIME *) pData);
    dbdatecrack(m_dbproc, &daterec, &datetime);
    m_txn.NewOrder.o_entry_d.year = daterec.year;
    m_txn.NewOrder.o_entry_d.month = daterec.month;
    m_txn.NewOrder.o_entry_d.day = daterec.day;
    m_txn.NewOrder.o_entry_d.hour = daterec.hour;
    m_txn.NewOrder.o_entry_d.minute = daterec.minute;
    m_txn.NewOrder.o_entry_d.second = daterec.second;
}
if (pData=dbdata(m_dbproc, 8))
    commit_flag = *(DBTINYINT *) pData);

DiscardNextRows(0);
DiscardNextResults(0);

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

```

```

        m_txn.NewOrder.exec_status_code = eInvalidItem;

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

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

```

```

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

    int          iTryCount = 0;
    const BYTE   *pData;

    ResetError();

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

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

```

```

&m_txn.Payment.c_id);

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

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

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

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

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

    if (pData=dbdata(m_dbproc, 1))
        m_txn.Payment.c_id = *((DBINT *) pData);
    if (pData=dbdata(m_dbproc, 2))
        UtilStrCpy(m_txn.Payment.c_last, pData, dbdatlen(m_dbproc, 2));
    if (pData=dbdata(m_dbproc, 3))
    {
        datetime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.Payment.h_date.year = daterec.year;
        m_txn.Payment.h_date.month = daterec.month;
        m_txn.Payment.h_date.day = daterec.day;
        m_txn.Payment.h_date.hour = daterec.hour;
        m_txn.Payment.h_date.minute = daterec.minute;
        m_txn.Payment.h_date.second = daterec.second;
    }
    if (pData=dbdata(m_dbproc, 4))
        UtilStrCpy(m_txn.Payment.w_street_1, pData, dbdatlen(m_dbproc,
4));
    if (pData=dbdata(m_dbproc, 5))
        UtilStrCpy(m_txn.Payment.w_street_2, pData, dbdatlen(m_dbproc,
5));
    if (pData=dbdata(m_dbproc, 6))
        UtilStrCpy(m_txn.Payment.w_city, pData, dbdatlen(m_dbproc, 6));
    if (pData=dbdata(m_dbproc, 7))
        UtilStrCpy(m_txn.Payment.w_state, pData, dbdatlen(m_dbproc, 7));
    if (pData=dbdata(m_dbproc, 8))
        UtilStrCpy(m_txn.Payment.w_zip, pData, dbdatlen(m_dbproc, 8));
    if (pData=dbdata(m_dbproc, 9))
        UtilStrCpy(m_txn.Payment.d_street_1, pData, dbdatlen(m_dbproc,
9));
    if (pData=dbdata(m_dbproc, 10))
        UtilStrCpy(m_txn.Payment.d_street_2, pData, dbdatlen(m_dbproc,
10));
    if (pData=dbdata(m_dbproc, 11))

```

```

        UtilStrCpy(m_txn.Payment.d_city, pData, dbdatlen(m_dbproc, 11));
if (pData=dbdata(m_dbproc, 12))
        UtilStrCpy(m_txn.Payment.d_state,  pData,  dbdatlen(m_dbproc,
12));
if (pData=dbdata(m_dbproc, 13))
        UtilStrCpy(m_txn.Payment.d_zip, pData, dbdatlen(m_dbproc, 13));
if (pData=dbdata(m_dbproc, 14))
        UtilStrCpy(m_txn.Payment.c_first, pData, dbdatlen(m_dbproc, 14));
if (pData=dbdata(m_dbproc, 15))
        UtilStrCpy(m_txn.Payment.c_middle,  pData,  dbdatlen(m_dbproc,
15));
if (pData=dbdata(m_dbproc, 16))
        UtilStrCpy(m_txn.Payment.c_street_1,  pData,  dbdatlen(m_dbproc,
16));
if (pData=dbdata(m_dbproc, 17))
        UtilStrCpy(m_txn.Payment.c_street_2,  pData,  dbdatlen(m_dbproc,
17));
if (pData=dbdata(m_dbproc, 18))
        UtilStrCpy(m_txn.Payment.c_city, pData, dbdatlen(m_dbproc, 18));
if (pData=dbdata(m_dbproc, 19))
        UtilStrCpy(m_txn.Payment.c_state, pData, dbdatlen(m_dbproc, 19));
if (pData=dbdata(m_dbproc, 20))
        UtilStrCpy(m_txn.Payment.c_zip, pData, dbdatlen(m_dbproc, 20));
if (pData=dbdata(m_dbproc, 21))
        UtilStrCpy(m_txn.Payment.c_phone,  pData,  dbdatlen(m_dbproc,
21));
if (pData=dbdata(m_dbproc, 22))
{
        datetime = *((DBDATETIME *) pData);
        dbdatecrack(m_dbproc, &daterec, &datetime);
        m_txn.Payment.c_since.year  = daterec.year;
        m_txn.Payment.c_since.month = daterec.month;
        m_txn.Payment.c_since.day   = daterec.day;
        m_txn.Payment.c_since.hour  = daterec.hour;
        m_txn.Payment.c_since.minute = daterec.minute;
        m_txn.Payment.c_since.second = daterec.second;
}
if(pData=dbdata(m_dbproc, 23))
        UtilStrCpy(m_txn.Payment.c_credit,  pData,  dbdatlen(m_dbproc,
23));
if(pData=dbdata(m_dbproc, 24))
        dbconvert(m_dbproc,  SQLNUMERIC,  (LPCBYTE)pData,
dbdatlen(m_dbproc,24), SQLFLT8, (BYTE *)&m_txn.Payment.c_credit_lim, 8);
if(pData=dbdata(m_dbproc, 25))
        dbconvert(m_dbproc,  SQLNUMERIC,  (LPCBYTE)pData,
dbdatlen(m_dbproc,25), SQLFLT8, (BYTE *)&m_txn.Payment.c_discount, 8);
if(pData=dbdata(m_dbproc, 26))
        dbconvert(m_dbproc,  SQLNUMERIC,  (LPCBYTE)pData,
dbdatlen(m_dbproc,26), SQLFLT8, (BYTE *)&m_txn.Payment.c_balance, 8);
if(pData=dbdata(m_dbproc, 27))
        UtilStrCpy(m_txn.Payment.c_data,  pData,  dbdatlen(m_dbproc, 27));

DiscardNextRows(0);

```

```

        DiscardNextResults(0);

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

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

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

```

```

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

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

    ResetError();

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

            dbrpcparam(m_dbproc, NULL, 0, SQLINT4, -1, -1, (BYTE *)

```



```

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

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

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

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

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

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

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

```

```

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

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

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

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

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

```

```

dbdatlen(m_dbproc,7),
                                SQLFLT8,                                (BYTE
*)&m_txn.OrderStatus.c_balance, 8);
                                if(pData=dbdata(m_dbproc, 8))
                                    m_txn.OrderStatus.o_id = (*(DBINT *) pData);

                                DiscardNextRows(0);
                                DiscardNextResults(0);

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

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

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

```

```

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

    ResetError();

    while (TRUE)
    {

```

```

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

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

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

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

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

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

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

            DiscardNextRows(0);
            DiscardNextResults(0);

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

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

```

```

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

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

```

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

};

class CDBLIBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eLogin,                // error from dblogin
        eDbOpen,              // error from dbopen
        eDbUse,                // error from dbuse
        eDbSqlExec,           // error from dbsqlexec
        eDbSet,                // error from one of the dbset* routines
        eDbNextRow,           // error from dbnextrow
        eWrongRowCount,       // more or less rows returned than
expected
        eWrongNumCols,        // more or less columns returned than
expected
        eDbResults,           // error from dbresults
        eDbRpcExec,           // error from dbrpcexec
        eDbSetMaxProcs,       // error from dbsetmaxprocs
        eDbProcHandler        // error from either dbprocerrhandle or
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;};
char*       ErrorTypeStr() { return "DBLIB"; }
int          ErrorNum() {return m_dberror;};
char*       ErrorText() {return m_dberrstr;};
int          ErrorAction() { return (int)m_eAction; }

};

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

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

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

    int          m_erno;
    int          m_iTryCount;

    int          ErrorType() {return ERR_TYPE_TPCC_DBLIB;};
    char*       ErrorTypeStr() { return "TPCC DBLIB"; }
    int          ErrorNum() {return m_erno;};

    char*       ErrorText();
};

```

```

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

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

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

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

        inline PNEW_ORDER_DATA BuffAddr_NewOrder() { return
&m_txn.NewOrder; };
        inline PPAYMENT_DATA BuffAddr_Payment() { return
&m_txn.Payment; };
        inline PDELIVERY_DATA BuffAddr_Delivery() { return
&m_txn.Delivery; };
        inline PSTOCK_LEVEL_DATA BuffAddr_StockLevel() {
&m_txn.StockLevel; };
        inline PORDER_STATUS_DATA BuffAddr_OrderStatus() {
&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. 4.42.000
 *          Copyright Microsoft, 2002
 *
 *      All Rights Reserved
 *
 *
 *          Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *
 *      PURPOSE:      Implements ODBC calls for TPC-C txns.
 *      Contact: Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *          4.42.000 - changed w_id fields from short to long to support >32K warehouses
 *          4.20.000 - updated rev number to match kit
 *          4.10.001 - not deleting error class in catch handler on deadlock retry;
 *                   not a functional bug, but a memory leak
 */

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

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

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

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

#ifdef ICECAP
#include <icapexp.h>

```

```

#endif

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

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

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

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

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

static SQLHENV henv = SQL_NULL_HENV; // ODBC environment
handle

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

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

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

/* FUNCTION: CTPCC_ODBC_ERR::ErrorText
*
*/

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

    static SERRORMSG errorMsgs[] =

```

```

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

```

```

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

```

```

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

```

```

// wrapper routine for class constructor

```

```

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

```

```

{
    return new CTPCC_ODBC( szServer, szUser, szPassword, szHost, szDatabase, szSPPrefix,
bCallNoDuplicatesNewOrder );
}

```

```

CTPCC_ODBC::CTPCC_ODBC (
    LPCSTR szServer,          // name of SQL server
    LPCSTR szUser,           // user name for login
    LPCSTR szPassword,       // password for login
    LPCSTR szHost,           // not used
    LPCSTR szDatabase,       // name of database to use
    LPCWSTR szSPPrefix,     // prefix to append to the stored procedure
names
    BOOL bCallNoDuplicatesNewOrder // whether to check for non-duplicate items in
NewOrder and call a new SP
)

```

```

: m_bCallNoDuplicatesNewOrder(bCallNoDuplicatesNewOrder)
{
    RETCODE          rc;

    // initialization
    m_hdbc = SQL_NULL_HDBC;
    m_hstmt = SQL_NULL_HSTMT;

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

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

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

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

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

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

#ifdef COMPILE_FOR_SNAC
        sprintf(
            szConnectStr,
            "DRIVER=SQL
Server;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
            szServer, szUser, szPassword, szDatabase );
#else
        // Compile for SNAC
        sprintf(
            szConnectStr,
            "DRIVER=SQL
Client;SERVER=%s;UID=%s;PWD=%s;DATABASE=%s",
            szServer, szUser, szPassword, szDatabase );
#endif
        rc = SQLDriverConnect(m_hdbc, NULL, (SQLCHAR*)szConnectStr,
            sizeof(szConnectStr),
            (SQLCHAR*)szOutStr, sizeof(szOutStr), &iOutStrLen,
            SQL_DRIVER_NOPROMPT );

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

    if (SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmt) != SQL_SUCCESS)

```

```

        ThrowError(CODBCERR::eAllocHandle);

    {
        char            buffer[128];

        // set some options affecting connection behavior
        strcpy(buffer, "set nocount on set XACT_ABORT ON");
        rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);

        // verify that version of stored procs on server is correct
        char db_sp_version[10];
        strcpy(buffer, "{call tpcc_version}");
        rc = SQLExecDirect(m_hstmt, (unsigned char *)buffer, SQL_NTS);
        if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)
            ThrowError(CODBCERR::eExecDirect);
        if ( SQLBindCol(m_hstmt, 1, SQL_C_CHAR, &db_sp_version,
sizeof(db_sp_version), NULL) != SQL_SUCCESS )
            ThrowError(CODBCERR::eBindCol);
        if ( SQLFetch(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eFetch);
        if (strcmp(db_sp_version,sVersion))
            throw
CTPCC_ODBC_ERR( CTPCC_ODBC_ERR::ERR_WRONG_SP_VERSION );
                new
                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          INativeError;
char            szState[6];
char            szMsg[SQL_MAX_MESSAGE_LENGTH];
char            szTmp[6*SQL_MAX_MESSAGE_LENGTH];
CODBCERR      *pODBCErr;          // not allocated until needed (maybe never)

pODBCErr = new CODBCERR();

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

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

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

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

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

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

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

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

SQLFreeStmt(m_hstmt, SQL_CLOSE);
throw pODBCErr;

```

```

}

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

    m_hstmt = m_hstmtStockLevel;

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

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

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

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

    m_hstmt = m_hstmtStockLevel;

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

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

            SQLFreeStmt(m_hstmt, SQL_CLOSE);

            m_txn.StockLevel.exec_status_code = eOK;
            break;
        }
    }
}

```

```

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

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

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

void CTPCC_ODBC::InitNewOrderParams()
{
    if (
        SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc, &m_hstmtNewOrder) !=
SQL_SUCCESS
        ||
        SQLAllocHandle(SQL_HANDLE_STMT, m_hdbc,
&m_hstmtNewOrderNoDuplicates) != SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols1) !=
SQL_SUCCESS
        || SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc, &m_descNewOrderCols2) !=
SQL_SUCCESS
        ||
        SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols1) != SQL_SUCCESS
        ||
        SQLAllocHandle(SQL_HANDLE_DESC, m_hdbc,
&m_descNewOrderNoDuplicatesCols2) != SQL_SUCCESS
    )
        ThrowError(CODBCERR::eAllocHandle);

    m_hstmt = m_hstmtNewOrder;

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

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

```



```

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

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

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

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

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

```

```

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

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

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

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

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

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

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

```

```

        ThrowError(CODBCERR::eBindParam);
    }

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

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

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

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

    //Compose the New Order statement
    _snwprintf(m_szNewOrderNoDuplicatesCommand,
    sizeof(m_szNewOrderNoDuplicatesCommand)/sizeof(m_szNewOrderNoDuplicatesCommand[0]),

```

```

L"{call %stpcc_neworder_new(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?"
L"?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)", m_szSPPrefix);

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

//
// Returns true if there are duplicate (warehouse_id, item_id)
// lineitem pairs in New Order input parameters.
//
bool CTPCC_ODBC::DuplicatesInNewOrder()
{
    int i, j;

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

    return false;
}

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

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

```

```

// 0      1
2
//
012345678901234567890123456789
    wchar_t          szSqlTemplate[iMAX_SP_NAME_LEN];
//=          L"{call
tpcc_neworder(?,?,?,?,"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
//
L"?,?,?,?,?,?,?,?,?,?,?,?,?}";

    m_hstmt = m_hstmtNewOrder;

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

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

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

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

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

```

```

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

        // move to the next resultset
        if ( SQLMoreResults(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eMoreResults);

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

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

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

    SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

//
// No lineitem duplicates optimized version.
//
void CTPCC_ODBC::NewOrderNoDuplicates()

```

```

{
    int                i;
    RETCODE           rc;
    int                iTryCount = 0;
                                // 0      1
2    3
                                //
0123456789012345678901234567890123
    wchar_t           szSqlTemplate[iMAX_SP_NAME_LEN];
                                // =      L" {call
tpcc_neworder_new(?,?,?,?,"
                                //
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
                                //
L"?,?,?,?,?,?,?,?,?,?,?,?,?"
                                //
L"?,?,?,?,?,?,?,?,?,?,?,?,?)"};

    m_hstmt = m_hstmtNewOrderNoDuplicates;

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

    // clip statement buffer based on number of parameters
    // fixed part is 33 chars and variable part is 6 chars per line item
    wcsncpy(szSqlTemplate, m_szNewOrderNoDuplicatesCommand);
    i = m_iBeginNewOrderNoDuplicatesVariablePart + m_txn.NewOrder.o_ol_cnt*6;
    wcsncpy( &szSqlTemplate[i], L" )" );

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

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

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

```

```

        ThrowError(CODBCERR::eExecDirect);

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

        // Get order line results
        if (SQLFetch(m_hstmt) == SQL_ERROR)
            ThrowError(CODBCERR::eFetch);

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

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

        // move to the next resultset
        if (SQLMoreResults(m_hstmt) == SQL_ERROR )
            ThrowError(CODBCERR::eMoreResults);

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

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

```

```

sizeof(m_txn.Payment.d_street_2), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.d_city,
sizeof(m_txn.Payment.d_city), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.d_state,
sizeof(m_txn.Payment.d_state), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.d_zip,
sizeof(m_txn.Payment.d_zip), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_first,
sizeof(m_txn.Payment.c_first), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_middle,
sizeof(m_txn.Payment.c_middle), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_street_1,
sizeof(m_txn.Payment.c_street_1), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_street_2,
sizeof(m_txn.Payment.c_street_2), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_city,
sizeof(m_txn.Payment.c_city), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_state,
sizeof(m_txn.Payment.c_state), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_zip,
sizeof(m_txn.Payment.c_zip), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_phone,
sizeof(m_txn.Payment.c_phone), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
&m_txn.Payment.c_since, 0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_credit,
sizeof(m_txn.Payment.c_credit), NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,  &m_txn.Payment.c_credit_lim,0,
NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,  &m_txn.Payment.c_discount,
0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE,  &m_txn.Payment.c_balance,
0, NULL) != SQL_SUCCESS
    || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR,    &m_txn.Payment.c_data,
sizeof(m_txn.Payment.c_data), NULL) != SQL_SUCCESS
)
    ThrowError(CODBCERR::eBindCol);

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

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

    m_hstmt = m_hstmtPayment;

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

```

```

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

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

        SQLFreeStmt(m_hstmt, SQL_CLOSE);

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

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

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

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

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

    m_hstmt = m_hstmtOrderStatus;

    if (
        SQLSetStmtAttrW(
            m_hstmt,
            SQL_ATTR_APP_ROW_DESC,
            m_descOrderStatusCols1, SQL_IS_POINTER ) != SQL_SUCCESS )

```

```

        ThrowError(CODBCERR::eSetStmtAttr);

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

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

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

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

        i = 0;
        if (
            SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_txn.OrderStatus.c_id, 0, NULL) !=
            SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_txn.OrderStatus.c_last,
            sizeof(m_txn.OrderStatus.c_last), NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_txn.OrderStatus.c_first,
            sizeof(m_txn.OrderStatus.c_first), NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_CHAR, &m_txn.OrderStatus.c_middle,
            sizeof(m_txn.OrderStatus.c_middle), NULL) != SQL_SUCCESS
            || SQLBindCol(m_hstmt, ++i, SQL_C_TYPE_TIMESTAMP,
            &m_txn.OrderStatus.o_entry_d, 0, NULL) != SQL_SUCCESS
        )

```

```

        || SQLBindCol(m_hstmt, ++i, SQL_C_SSHORT, &m_txn.OrderStatus.o_carrier_id,
0, NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_DOUBLE, &m_txn.OrderStatus.c_balance, 0,
NULL) != SQL_SUCCESS
        || SQLBindCol(m_hstmt, ++i, SQL_C_SLONG, &m_txn.OrderStatus.o_id, 0,
NULL) != SQL_SUCCESS
    )
    ThrowError(CODBCERR::eBindCol);

    //Compose Order Status statement
    _snwprintf(m_szOrderStatusCommand,
sizeof(m_szOrderStatusCommand)/sizeof(m_szOrderStatusCommand[0]),
    L" {call %stpcc_orderstatus (?,?,?,?)}", m_szSPPrefix);
}

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

    m_hstmt = m_hstmtOrderStatus;

    if ( SQLSetStmtAttrW( m_hstmt, SQL_ATTR_APP_ROW_DESC, m_descOrderStatusCols1,
SQL_IS_POINTER ) != SQL_SUCCESS )
        ThrowError(CODBCERR::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, m_szOrderStatusCommand, 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_SLONG,
SQL_INTEGER, 0, 0, &m_txn.Delivery.w_id, 0, NULL) != SQL_SUCCESS
            || SQLBindParameter(m_hstmt, ++i, SQL_PARAM_INPUT, SQL_C_SSHORT,
SQL_SMALLINT, 0, 0, &m_txn.Delivery.o_carrier_id, 0, NULL) != SQL_SUCCESS
        )
            ThrowError(CODBCERR::eBindParam);

        for (i=0;i<10;i++)
        {
            if (
                SQLBindCol(m_hstmt, (UWORD)(i+1), SQL_C_SLONG,
&m_txn.Delivery.o_id[i], 0, NULL) != SQL_SUCCESS )
                ThrowError(CODBCERR::eBindCol);
        }

        //Compose Delivery statement
        _snwprintf(m_szDeliveryCommand,
sizeof(m_szDeliveryCommand)/sizeof(m_szDeliveryCommand[0]),
            L"{call %stpcc_delivery (?,?)}", m_szSPPrefix);
    }

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

    m_hstmt = m_hstmtDelivery;

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

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

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

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

```

```

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

```

tpcc_odbc.h

```

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

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

#define iMAX_SP_NAME_LEN 256 //maximum length of a stored procedure name with
parameters

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

```



```

        eSetStmtAttr          // error from SQLSetStmtAttr
    };

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

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

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

    int       ErrorType() {return ERR_TYPE_ODBC;};
    char*     ErrorTypeStr() { return "ODBC"; }
    int       ErrorNum() {return m_NativeError;};
    char*     ErrorText() {return m_odbcerrstr;};
    int       ErrorAction() { return (int)m_eAction; }
};

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

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

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

    int       m_erno;
    int       m_iTryCount;

    int       ErrorType() {return ERR_TYPE_TPCC_ODBC;};
    char*     ErrorTypeStr() { return "TPCC ODBC"; }
};

```

```

        int            ErrorNum() {return m_errno;};

        char*   ErrorText();
};

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

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

        SQLHSTMT       m_hstmtNewOrder;
        SQLHSTMT       m_hstmtNewOrderNoDuplicates; // NewOrder with one result set
for lineitem details
        SQLHSTMT       m_hstmtPayment;
        SQLHSTMT       m_hstmtDelivery;
        SQLHSTMT       m_hstmtOrderStatus;
        SQLHSTMT       m_hstmtStockLevel;

        SQLHDESC       m_descNewOrderCols1;
        SQLHDESC       m_descNewOrderCols2;
        SQLHDESC       m_descNewOrderNoDuplicatesCols1; // NewOrder with one
result set for lineitem details
        SQLHDESC       m_descNewOrderNoDuplicatesCols2; // NewOrder with one
result set for lineitem details
        SQLHDESC       m_descOrderStatusCols1;
        SQLHDESC       m_descOrderStatusCols2;

        wchar_t        m_szSPPrefix[32];    // stored procedures prefix

        wchar_t        m_szNewOrderCommand[iMAX_SP_NAME_LEN];
        wchar_t        m_szNewOrderNoDuplicatesCommand[iMAX_SP_NAME_LEN];
        int            m_iBeginNewOrderVariablePart; // beginning of the
variable part in NewOrder statement
        int            m_iBeginNewOrderNoDuplicatesVariablePart; //
begining of the variable part in NewOrder statement
        wchar_t        m_szPaymentCommand[iMAX_SP_NAME_LEN];
        wchar_t        m_szDeliveryCommand[iMAX_SP_NAME_LEN];
        wchar_t        m_szOrderStatusCommand[iMAX_SP_NAME_LEN];
        wchar_t        m_szStockLevelCommand[iMAX_SP_NAME_LEN];

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

```

```

// tpcc_neworder_new flag
BOOL          m_bCallNoDuplicatesNewOrder;

void ThrowError( CODBCERR::ACTION eAction );

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

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

bool DuplicatesInNewOrder();
void NewOrderDuplicates();
void NewOrderNoDuplicates();

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

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

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

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_ODBC* CTPCC_ODBC_new

```

```
( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword,
  LPCSTR szHost, LPCSTR szDatabase,
  LPCWSTR szSPPrefix, BOOL bCallNoDuplicatesNewOrder );
```

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

tpcc_oledb.cpp

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

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

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

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

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

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

#ifdef SQL_MAX_MESSAGE_LENGTH
#define SQL_MAX_MESSAGE_LENGTH 512
#endif

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

```

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

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

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

BOOL WINAPI DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:
            DisableThreadLibraryCalls(hModule);
            break;

        case DLL_PROCESS_DETACH:
            break;

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

/* FUNCTION: CTPCC_OLEDB_ERR::ErrorText
 *
 */

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

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

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

    for(i=0; errorMsgs[i].szMsg[0]; i++)

```

```

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

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

CTPCC_OLEDB::CTPCC_OLEDB (
    LPCSTR szServer,           // name of SQL server
    LPCSTR szUser,            // user name for login
    LPCSTR szPassword,        // password for login
    LPCSTR szHost,            // not used
    LPCSTR szDatabase,        // name of database to use
    LPCWSTR szSPPrefix        // prefix to append to the stored procedure
names
)
: m_pIMalloc(NULL)
{
    int                iRc;
    int                i;
    HRESULT             hr;
    IDBInitialize*    pIDBInitialize = NULL;           // data source interface
    IDBProperties*     pIDBProperties = NULL;
    ICommandText*     pICommandText;                 // SQL command
without parameters
    wchar_t            szwServer[iMaxNameLen];       // Unicode string
used to convert to BSTR
    wchar_t            szwDatabase[iMaxNameLen];    // Unicode string used to
convert to BSTR
    wchar_t            szwUser[iMaxNameLen];        // Unicode string
used to convert to BSTR
    wchar_t            szwPassword[iMaxNameLen];    // Unicode string used to
convert to BSTR

    // Copy stored procedures prefix
    wcsncpy(m_szSPPrefix, szSPPrefix, sizeof(m_szSPPrefix)/sizeof(m_szSPPrefix[0]));
}

```

```

        // Convert single byte ANSI strings to Unicode (for later conversion to BSTR)
        iRc = MultiByteToWideChar(CP_THREAD_ACP, MB_PRECOMPOSED, szServer,
(int)strlen(szServer)+1, szwServer, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP, MB_PRECOMPOSED, szDatabase,
(int)strlen(szDatabase)+1, szwDatabase, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP, MB_PRECOMPOSED, szUser,
(int)strlen(szUser)+1, szwUser, iMaxNameLen);
        iRc = MultiByteToWideChar(CP_THREAD_ACP, MB_PRECOMPOSED, szPassword,
(int)strlen(szPassword)+1, szwPassword, iMaxNameLen);

        // Initialize COM library to be able to use OLE-DB interfaces
        CoInitialize(NULL);

        // Initialization - create SQLOLEDB component
        //hr = CoCreateInstance(CLSID_SQLOLEDB, // GUID of SQLOLEDB component
        // Compile for SNAC
        hr = CoCreateInstance(CLSID_SQLNCLI, // GUID of SQLNCLI component
        NULL, // not defining an aggregate component, so NULL
        CLSCTX_INPROC_SERVER, // run the component in our process
        IID_IDBInitialize,
        (void **) &pIDBInitialize);
/*
Initialize the property values needed
to establish the connection.
*/
for(i = 0; i < 4; i++)
    VariantInit(&m_InitProperties[i].vValue);
//Server name.
m_InitProperties[0].dwPropertyID = DBPROP_INIT_DATASOURCE;
m_InitProperties[0].vValue.vt = VT_BSTR;
m_InitProperties[0].vValue.bstrVal= SysAllocString(szwServer);
m_InitProperties[0].dwOptions = DBPROPOPTIONS_REQUIRED;
m_InitProperties[0].colid = DB_NULLID;
//Database.
m_InitProperties[1].dwPropertyID = DBPROP_INIT_CATALOG;
m_InitProperties[1].vValue.vt = VT_BSTR;
m_InitProperties[1].vValue.bstrVal= SysAllocString(szwDatabase);
m_InitProperties[1].dwOptions = DBPROPOPTIONS_REQUIRED;
m_InitProperties[1].colid = DB_NULLID;
//Username (login).
m_InitProperties[2].dwPropertyID = DBPROP_AUTH_USERID;
m_InitProperties[2].vValue.vt = VT_BSTR;
m_InitProperties[2].vValue.bstrVal= SysAllocString(szwUser);
m_InitProperties[2].dwOptions = DBPROPOPTIONS_REQUIRED;
m_InitProperties[2].colid = DB_NULLID;
//Password.
m_InitProperties[3].dwPropertyID = DBPROP_AUTH_PASSWORD;
m_InitProperties[3].vValue.vt = VT_BSTR;
m_InitProperties[3].vValue.bstrVal= SysAllocString(szwPassword);
m_InitProperties[3].dwOptions = DBPROPOPTIONS_REQUIRED;
m_InitProperties[3].colid = DB_NULLID;
/*
Construct the DBPROPSET structure(m_rgInitPropSet). The

```

```

DBPROPSET structure is used to pass an array of DBPROP
structures (m_InitProperties) to the SetProperty method.
*/
m_rgInitPropSet.guidPropertySet = DBPROPSET_DBINIT;
m_rgInitPropSet.cProperties = 4;
m_rgInitPropSet.rgProperties = m_InitProperties;
//Set initialization properties.
    if (FAILED(hr = pIDBInitialize->QueryInterface(IID_IDBProperties,
        (void **)&pIDBProperties)))
    {
        ThrowError(pIDBInitialize, COLEDBERR::eQueryInterface, "CTPCC_OLEDB()");
    }

hr = pIDBProperties->SetProperties(1, &m_rgInitPropSet);

pIDBProperties->Release();
//Now establish the connection to the data source.
hr = pIDBInitialize->Initialize();

    // Free BSTR property strings
    for(i = 0; i < 4; i++)
    {
        SysFreeString(m_InitProperties[i].vValue.bstrVal);
    }

    hr = pIDBInitialize->QueryInterface(IID_IDBCreateSession, (void
***)&m_pIDBCreateSession);

    // Releasing this has no effect on the SQL Server connection
// of the data source object because of the reference maintained by
// m_pIDBCreateSession.
pIDBInitialize->Release();
pIDBInitialize = NULL;

    hr = m_pIDBCreateSession->CreateSession(NULL, IID_IDBCreateCommand, (IUnknown
***)&m_pIDBCreateCommand);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateSession, COLEDBERR::eCreateSession,
"CTPCC_OLEDB()");
    }

    hr = m_pIDBCreateCommand->CreateCommand(NULL, IID_ICommandText, (IUnknown
***)&pICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand, COLEDBERR::eCreateCommand,
"CTPCC_OLEDB()");
    }

    hr = pICommandText->SetCommandText(DBGUID_SQL, L"set nocount on set
XACT_ABORT ON");

```



```

        if (FAILED(hr))
        {
            ThrowError(pICommandText, COLEDBERR::eSetCommandText,
"CTPCC_OLEDB()");
        }

        hr = pICommandText->Execute(NULL, IID_NULL, NULL, NULL, NULL);
        if (FAILED(hr))
        {
            ThrowError(pICommandText, COLEDBERR::eExecute, "CTPCC_OLEDB()");
        }

        pICommandText->Release();

        // verify that version of stored procs on server is correct
        CheckSPVersion();

        // Get IMalloc interface
        hr = CoGetMalloc(1, (LPMALLOC *)&m_pIMalloc);

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

CTPCC_OLEDB::~CTPCC_OLEDB( void )
{
    if (m_pIMalloc != NULL)
    {
        m_pIMalloc->Release();
    }
    m_pIPaymentCommand->Release();
    m_pIDBCreateCommand->Release();
    m_pIDBCreateSession->Release();

    CoUninitialize();// uninitialized COM library
}

/*
 *      Check stored procedures version on the server.
 */
void CTPCC_OLEDB::CheckSPVersion()
{
    HRESULT          hr;
    char             db_sp_version[10];
    ICommandText*   pICommandText;
    IAccessor*       pIAccessor;
    IRowset*         pRowset;
    const ULONG     nOutputParams = 1;    // output 1st result set columns
    HACCESSOR        hTpcVersionOutputAccessor;

```

```

// Structure to bind in accessor
DBBINDING          acOutputDBBinding[nOutputParams];
DBBINDSTATUS       acOutputDBBindStatus[nOutputParams];
LONG               cRows = 1;      // number of rows returned in the rowset
ULONG              cRowsObtained;
HROW               rghRow;         //returned row handles
HROW*              prghRow = &rghRow;

    hr = m_pIDBCreateCommand->CreateCommand(NULL, IID ICommandText, (IUnknown
**) &pICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand,          COLEDBERR::eCreateCommand,
"CheckSPVersion()");
    }

    hr = pICommandText->SetCommandText(DBGUID_SQL, L"{call tpcc_version}");
    if (FAILED(hr))
    {
        ThrowError(pICommandText,              COLEDBERR::eSetCommandText,
"CheckSPVersion()");
    }

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

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

// Binding for a rowset
SetBinding(&acOutputDBBinding[0], 0, sizeof(db_sp_version), DBTYPE_STR);

hr = pIAccessor->CreateAccessor(
    DBACCESSOR_ROWDATA,
    nOutputParams,
    acOutputDBBinding,
    sizeof(db_sp_version),
    &hTpccVersionOutputAccessor,
    acOutputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "CheckSPVersion()");
}

hr = pICommandText->Execute(NULL, IID_IRowset, NULL, NULL, (IUnknown
**) &pRowset);
if (FAILED(hr))
{

```

```

        ThrowError(pICommandText, COLEDBERR::eExecute, "CheckSPVersion()");
    }

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

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

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

    pICommandText->Release();

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

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

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

```

```

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

assert(pObjectWithError != NULL);

pOLEDBErr = new COLEDBERR(szLocation);

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

szTmp[0] = 0;

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

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

if (pIErrorInfoAll != NULL)
{
    // Test to see if it's a valid OLE DB IErrorInfo interface
    // exposing a list of records.
    if (SUCCEEDED(pIErrorInfoAll->QueryInterface(IID_IErrorRecords,      (void**)
&pIErrorRecords)))
    {
        pIErrorRecords->GetRecordCount(&nRecs);

        // Within each record, retrieve information from each
        // of the defined interfaces.
        for (nRec = 0; nRec < nRecs; nRec++)
        {

```

```

// Request the generic SQL error interface.
pLErrorRecords->GetCustomErrorObject(nRec,

        IID_ISQLErrorInfo, // generic SQL error interface
        (IUnknown**) &pISQLErrorInfo);

if (pISQLErrorInfo != NULL)
{
    // Request SQL Server-specific error interface, not the
generic SQL error interface.
    pISQLErrorInfo->QueryInterface(
        IID_ISQLServerErrorInfo, // SQL Server error
interface
        (void**) &pISQLServerErrorInfo);
}

// Test to ensure the reference is valid, then
// get error information from ISQLServerErrorInfo.
if (pISQLServerErrorInfo != NULL)
{
    pISQLServerErrorInfo->GetErrorInfo(&pSSErrorInfo,
&pSSErrorStrings);

    // ISQLServerErrorInfo::GetErrorInfo succeeds
    // even when it has nothing to return. Test the
    // pointers before using.
    if (pSSErrorInfo)
    {
        // First, add the error message.

        // Convert Unicode error string to ANSI.
        WideCharToMultiByte(CP_THREAD_ACP, 0,
            pSSErrorInfo->pwszMessage, -1,
            szMsg, sizeof(szMsg),
            NULL, NULL);

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

        // include line break after first error msg
        if (szTmp[0] != 0)
            strcat( szTmp, "\r\n");
        // concatenate the error record to the overall error
message
        strcat( szTmp, szMsg );

        // Second, add the stored procedure name and line
number, if available.
        if (wcslen(pSSErrorInfo->pwszProcedure)>0)

```

```

    {
        // Prefix with a line break
        iLen = sprintf(szMsg, "\r\nProcedure: ");

        // Convert Unicode error string to ANSI.
        WideCharToMultiByte(CP_THREAD_ACP, 0,
            pSSErrorInfo->pwszProcedure, -
            1,
            &szMsg[iLen], sizeof(szMsg) -
            iLen,
            NULL, NULL);

        // Check if have space to add the line
        // Assume the line number takes no more
        // than 3 digits.
        if ((strlen(szMsg) + 4) < sizeof(szMsg))
        {
            _snprintf(&szMsg[strlen(szMsg)],
                sizeof(szMsg),
                "%d", pSSErrorInfo->
                wLineNumber);
        }

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

        // concatenate the error record to the
        // overall error message
        strcat( szTmp, szMsg );

        // copy the overall error string to the
        // exception
        pOLEDBErr->m_OLEDBErrStr = new
        char[strlen(szTmp)+1];
        strcpy(pOLEDBErr->m_OLEDBErrStr,
            szTmp);
    }

    // Third, capture the (first) database error
    if (pOLEDBErr->m_NativeError == 0 &&
        pSSErrorInfo->INative != 0)
    {
        pOLEDBErr->m_NativeError =
        pSSErrorInfo->INative;

        // Check for deadlock error code and set
        // the deadlock flag
    }

```

```

        if (pSSErrorInfo->INative == 1205)
        {
            pOLEDBErr->m_bDeadLock =
TRUE;
        }
    }

    // IMalloc::Free needed to release references
    // on returned values.
    if (m_pIMalloc != NULL)
    {
        m_pIMalloc->Free(pSSErrorStrings);
        m_pIMalloc->Free(pSSErrorInfo);
    }
}

pISQLServerErrorInfo->Release();
}
else
{
    // Custom error object is not supported.
    // Use general OLE-DB error interface.

    // Get the numeric error code
    pErrorRecords->GetBasicErrorInfo(nRec,
&BasicErrorInfo);

    if (pOLEDBErr->m_NativeError == 0)
    {
        // Get the failed call HRESULT code, which is not
really the native error
        pOLEDBErr->m_NativeError =
BasicErrorInfo.hrError;
    }

    // Try to get the string description of the error.
    pErrorRecords->GetErrorInfo(nRec,
LOCALE_USER_DEFAULT, (IErrorInfo**)&pErrorInfoRecord);

    if (pErrorInfoRecord)
    {
        pErrorInfoRecord->GetDescription(&bstrDescription);

        // Convert Unicode error string to ANSI.
        WideCharToMultiByte(CP_THREAD_ACP, 0,
            bstrDescription, -1,
            szMsg, sizeof(szMsg),
            NULL, NULL);

        pOLEDBErr->m_OLEDBErrStr = new
char[strlen(szMsg)+1];
        strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
    }
}
}
}

```

```

        }
    } // for()
} // if (SUCCEEDED(pIErrorInfoAll->QueryInterface(IID_IErrorRecords,
(void**) &pIErrorRecords)))
else
{
    // No IErrorRecords interface supported. Use default IErrorInfo.
    // Note: SQLOLEDB supports IErrorRecords, so this check is for good style
only.
    _snprintf(szMsg, sizeof(szMsg), "IErrorRecords interface not supported");

    pOLEDBErr->m_OLEDBErrStr = new char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}

pIErrorInfoAll->Release();

} // if (pIErrorInfoAll != NULL)
else
{
    // No IErrorInfo interface supported.
    // Note: SQLOLEDB supports IErrorInfo, so this check is for good style only.
    _snprintf(szMsg, sizeof(szMsg), "IErrorInfo interface not supported");
    pOLEDBErr->m_OLEDBErrStr = new char[strlen(szMsg)+1];
    strcpy(pOLEDBErr->m_OLEDBErrStr, szMsg);
}

throw pOLEDBErr;
}

/*
 *
 * Create a new command object from the SQL text passed in.
 *
 */
void CTPCC_OLEDB::CreateCommand(wchar_t* szSQLCommand, // I:
SQL query for the command
                                ICommandText**
ppICommandText // O: returned command object
                                )
{
    HRESULT hr;

    // Create a new command object
    hr = m_pIDBCreateCommand->CreateCommand(NULL, IID_ICommandText, (IUnknown
**)ppICommandText);
    if (FAILED(hr))
    {
        ThrowError(m_pIDBCreateCommand, COLEDBERR::eCreateCommand,
"CTPCC_OLEDB::CreateCommand");
    }
}

```



```

// Set command text
hr = (*ppICommandText)->SetCommandText(DBGUID_SQL, szSQLCommand);
if (FAILED(hr))
{
    ThrowError(*ppICommandText, COLEDBERR::eSetCommandText,
"CTPCC_OLEDB::CreateCommand");
}

// Prepare the command
PrepareCommand(*ppICommandText);
}

/*
* QueryInterface and Prepare in one function for simplicity.
* DEFERRED PREPARE property is set to off to prepare immediatelly.
*/
void CTPCC_OLEDB::PrepareCommand(ICommandText* pICommandText)
{
    HRESULT hr;
    ICommandPrepare* pICommandPrepare;
    ICommandProperties* pICommandProperties;
    DBPROPSET rowSetPropSet;
    DBPROP rowSetProp;

// Set the deferred prepare property to false.
rowSetProp.dwPropertyID = SSPROP_DEFERPREPARE;
memset(&rowSetProp.vValue, 0, sizeof(rowSetProp.vValue));
rowSetProp.dwOptions = DBPROPOPTIONS_REQUIRED;
rowSetProp.colid = DB_NULLID;

rowSetPropSet.cProperties = 1;
rowSetPropSet.guidPropertySet = DBPROPSET_SQLSERVERROWSET;
rowSetPropSet.rgProperties = &rowSetProp;

// Query interface for setting properties
hr = pICommandText->QueryInterface(IID_ICommandProperties, (void
**) &pICommandProperties);
if (FAILED(hr))
{
    ThrowError(pICommandText, COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
}

// Set the property set
hr = pICommandProperties->SetProperties(1, &rowSetPropSet);
if (FAILED(hr))
{
    ThrowError(pICommandText, COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
}

// Get interface for preparing commands
hr = pICommandText->QueryInterface(IID_ICommandPrepare, (void

```

```

**) &pICommandPrepare);
    if (FAILED(hr))
    {
        ThrowError(pICommandText, COLEDBERR::eQueryInterface,
"CTPCC_OLEDB::PrepareCommand");
    }

    // Prepare Payment command
    hr = pICommandPrepare->Prepare(0xFFFFFFFF);
    if (FAILED(hr))
    {
        ThrowError(pICommandPrepare, COLEDBERR::ePrepare,
"CTPCC_OLEDB::PrepareCommand");
    }
}

/*
 * Initialize fields of an array of bindings structures.
 * Needs to be called before setting individual parameter/column bindings.
 */
void CTPCC_OLEDB::InitBindings(DBBINDING* pDBBindings, // IO: array of bindings
                               int iCount,
                               // I: number of elements in the array
                               eBindingType BindingType) // I:
what the bindings will be used for (parameters/columns)
{
    int i;

    for(i = 0; i < iCount; i++)
    {
        pDBBindings[i].iOrdinal = i + 1;
        pDBBindings[i].obLength = 0;
        pDBBindings[i].obStatus = 0;
        pDBBindings[i].pTypeInfo = NULL;
        pDBBindings[i].pObject = NULL;
        pDBBindings[i].pBindExt = NULL;
        pDBBindings[i].dwPart = DBPART_VALUE;

        switch (BindingType)
        {
            case eInputParameter:
                pDBBindings[i].eParamIO = DBPARAMIO_INPUT;
                break;
            case eOutputParameter:
                pDBBindings[i].eParamIO = DBPARAMIO_OUTPUT;
                break;
            case eInputOutputParameter:
                pDBBindings[i].eParamIO = DBPARAMIO_INPUT |
DBPARAMIO_OUTPUT;
                break;
            case eOutputColumn:
                pDBBindings[i].eParamIO = DBPARAMIO_NOTPARAM;
                break;
        }
    }
}

```

```

        default:
            assert(false);    // this should never happen
        }

    pDBBindings[i].dwMemOwner = DBMEMOWNER_CLIENTOWNED;
    pDBBindings[i].dwFlags = 0;
        pDBBindings[i].bPrecision = 0;
    pDBBindings[i].bScale = 0;
    }
}

/*
 * Perform binding for one parameter or output column.
 *
 */
void CTPCC_OLEDB::SetBinding(DBBINDING* pDBBinding, // I: binding row structure
                             size_t obValue,        // I:
                             parameter (column) offset in the user buffer
                             size_t cbMaxLen,        // I:
                             parameter (column) length
                             DBTYPE wType
                             // I: parameter (column) type
                             )
{
    pDBBinding->obValue = (ULONG)obValue;
    pDBBinding->cbMaxLen = (ULONG)cbMaxLen;
    pDBBinding->wType = wType;
}

void CTPCC_OLEDB::InitStockLevelParams()
{
    int i;
    HRESULT hr;
    wchar_t szName[iMAX_SP_NAME_LEN];
    IAccessor* pIAccessor;
    const ULONG nInputParams = 3; // input parameters
    const ULONG nOutputParams = 1; // output 1st result set
    columns
    // Structure to bind in accessor
    DBBINDING acInputDBBinding[nInputParams];
    DBBINDSTATUS acInputDBBindStatus[nInputParams];
    DBBINDING acOutputDBBinding[nOutputParams];
    DBBINDSTATUS acOutputDBBindStatus[nOutputParams];

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

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

    // Describe the consumer buffer by filling in the array
    // of DBBINDING structures. Each binding associates

```

```

// a single parameter to the consumer's buffer.
    InitBindings(&acInputDBBinding[0], nInputParams, eInputParameter);

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

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

// StockLevel parameter 3
    SetBinding(&acInputDBBinding[i++],      offsetof(STOCK_LEVEL_DATA,      threshold),
sizeof(m_txn.StockLevel.threshold), DBTYPE_I2);

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

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_PARAMETERDATA,
        nInputParams,
        acInputDBBinding,
        sizeof(STOCK_LEVEL_DATA),
        &m_hStockLevelInputAccessor,
        acInputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor,      COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }

    m_StockLevelExecuteParams.cParamSets = 1;
    m_StockLevelExecuteParams.hAccessor = m_hStockLevelInputAccessor;
    m_StockLevelExecuteParams.pData = &m_txn.StockLevel;

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

// Binding for a rowset that may return more than one row.
    i = 0;
// StockLevel output column 1
    SetBinding(&acOutputDBBinding[i++],      offsetof(STOCK_LEVEL_DATA,      low_stock),
sizeof(m_txn.StockLevel.low_stock), DBTYPE_I4);

    hr = pIAccessor->CreateAccessor(
        DBACCESSOR_ROWDATA | DBACCESSOR_OPTIMIZED,
        nOutputParams,
        acOutputDBBinding,

```

```

        sizeof(STOCK_LEVEL_DATA),
        &m_hStockLevelOutputAccessor,
        acOutputDBBindStatus);
    if (FAILED(hr))
    {
        ThrowError(pIAccessor, COLEDBERR::eCreateAccessor,
"InitStockLevelParams()");
    }
}

void CTPCC_OLEDB::StockLevel()
{
    HRESULT          hr;
    int              iTryCount = 0;
    IRowset*        pRowset;
    LONG            cRows = 1;    // number of rows returned in the rowset
    ULONG           cRowsObtained;
    HROW            rghRow;      //returned row handles
    HROW*           prghRow = &rghRow;

    while (TRUE)
    {
        try
        {
            // Execute the prepared command
            hr = m_pIStockLevelCommand->Execute(NULL, IID_IRowset,
&m_StockLevelExecuteParams, NULL,

(IUnknown **)&pRowset);
            if (FAILED(hr))
            {
                ThrowError(m_pIStockLevelCommand, COLEDBERR::eExecute,
"StockLevel()");
            }

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

            // Fetch the actual row data by handle
            hr = pRowset->GetData(rghRow, m_hStockLevelOutputAccessor,
&m_txn.StockLevel);
            if (FAILED(hr))
            {
                ThrowError(m_pIStockLevelCommand, COLEDBERR::eGetData,
"StockLevel()");
            }
        }
    }
}

```

```

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

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

        m_txn.StockLevel.exec_status_code = eOK;

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

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

//      if (iTryCount)
//      throw
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

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

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

    i = 0;
    // NewOrder parameter 1

```

```

        SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,      w_id),
sizeof(m_txn.NewOrder.w_id), DBTYPE_I4);

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

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

// NewOrder parameter 4
        SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,      o_ol_cnt),
sizeof(m_txn.NewOrder.o_ol_cnt), DBTYPE_UI1);

// NewOrder parameter 5
        SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,      o_all_local),
sizeof(m_txn.NewOrder.o_all_local), DBTYPE_UI1);

        for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
                SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,
OL[j].ol_i_id), sizeof(m_txn.NewOrder.OL[j].ol_i_id), DBTYPE_I4);
                SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,
OL[j].ol_supply_w_id), sizeof(m_txn.NewOrder.OL[j].ol_supply_w_id), DBTYPE_I4);
                SetBinding(&acInputDBBinding[i++],      offsetof(NEW_ORDER_DATA,
OL[j].ol_quantity), sizeof(m_txn.NewOrder.OL[j].ol_quantity), DBTYPE_I2);
        }

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

// Binding for the order line rowsets (each consist of one row).
// Bind to offsets of the OL_NEW_ORDER_DATA structure instead of
NEW_ORDER_DATA.
// IRowset::GetData() will be passed individual array slots OL[i] to fetch the data
// from the row set.

        i = 0;
// NewOrder output column 1
        SetBinding(&acOutputDBBinding[i++],      offsetof(OL_NEW_ORDER_DATA,      ol_i_name),
sizeof(m_txn.NewOrder.OL[0].ol_i_name), DBTYPE_STR);

// NewOrder output column 2
        SetBinding(&acOutputDBBinding[i++],      offsetof(OL_NEW_ORDER_DATA,      ol_stock),
sizeof(m_txn.NewOrder.OL[0].ol_stock), DBTYPE_I2);

// NewOrder output column 3
        SetBinding(&acOutputDBBinding[i++],      offsetof(OL_NEW_ORDER_DATA,
ol_brand_generic), sizeof(m_txn.NewOrder.OL[0].ol_brand_generic), DBTYPE_STR);

// NewOrder output column 4

```

```

        SetBinding(&acOutputDBBinding[i++],  offsetof(OL_NEW_ORDER_DATA,  ol_i_price),
sizeof(m_txn.NewOrder.OL[0].ol_i_price), DBTYPE_R8);

        // NewOrder output column 5
        SetBinding(&acOutputDBBinding[i++],  offsetof(OL_NEW_ORDER_DATA,  ol_amount),
sizeof(m_txn.NewOrder.OL[0].ol_amount), DBTYPE_R8);

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

        i = 0;
        // NewOrder output column 1
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      w_tax),
sizeof(m_txn.NewOrder.w_tax), DBTYPE_R8);

        // NewOrder output column 2
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      d_tax),
sizeof(m_txn.NewOrder.d_tax), DBTYPE_R8);

        // NewOrder output column 3
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      o_id),
sizeof(m_txn.NewOrder.o_id), DBTYPE_I4);

        // NewOrder output column 4
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      c_last),
sizeof(m_txn.NewOrder.c_last), DBTYPE_STR);

        // NewOrder output column 5
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      c_discount),
sizeof(m_txn.NewOrder.c_discount), DBTYPE_R8);

        // NewOrder output column 6
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      c_credit),
sizeof(m_txn.NewOrder.c_credit), DBTYPE_STR);

        // NewOrder output column 7
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      o_entry_d),
sizeof(m_txn.NewOrder.o_entry_d), DBTYPE_DBTIMESTAMP);

        // NewOrder output column 8
        SetBinding(&acOutputDBBinding2[i++],  offsetof(NEW_ORDER_DATA,      o_commit_flag),
sizeof(m_txn.NewOrder.o_commit_flag), DBTYPE_I2);

        for (j=0; j<MAX_OL_NEW_ORDER_ITEMS; j++)
        {
            // Set command text first

            // Print the fixed first portion of parameters
            i = _snwprintf(szName, sizeof(szName)/sizeof(szName[0]),
                L"{call %stpcc_neworder (?,?,?,?,?,?", m_szSPPrefix);

            // Now print the variable portion depending on the number of order line parameters

```



```

for (iOlCount = 0; iOlCount <= j; ++iOlCount)
{
    i += _snwprintf(&szName[i], sizeof(szName)/sizeof(szName[0]) - i,
L",?,?,?");

}

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

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

// Now create the input accessor for this prepared command
hr = m_pINewOrderCommand[j]->QueryInterface(IID_IAccessor, (void
**)&pIAccessor);
if (FAILED(hr))
{
    ThrowError(m_pINewOrderCommand[j], COLEDBERR::eQueryInterface,
"InitNewOrderParams()");
}

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

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

```

```

m_NewOrderExecuteParams[j].pData = &m_txn.NewOrder;

// Create accessor for the first rowset
hr = pIAccessor->CreateAccessor(
    DBACCESSOR_ROWDATA | DBACCESSOR_OPTIMIZED,
    nOutputParams,
    acOutputDBBinding,
    sizeof(OL_NEW_ORDER_DATA),
    &m_hNewOrderOutputAccessor[j],
    acOutputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor,          COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
}

// Create accessor for the second rowset
hr = pIAccessor->CreateAccessor(
    DBACCESSOR_ROWDATA, // cannot be optimized too because #1
accessor is
    nOutputParams2,
    acOutputDBBinding2,
    sizeof(NEW_ORDER_DATA),
    &m_hNewOrderOutputAccessor2[j],
    acOutputDBBindStatus2);
if (FAILED(hr))
{
    ThrowError(pIAccessor,          COLEDBERR::eCreateAccessor,
"InitNewOrderParams()");
}

    pIAccessor->Release();
}
}

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

```

```

HROW*                prghRows2 = &rgHRows2;
int                  i;
long                 IRowsAffected; // the number of affected rows for a rowset
int                  iHandleIndex; // index into the handle arrays
based on the orders count

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

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

while (TRUE)
{
    try
    {
        // Execute the prepared command (according to the number of new orders)
        // Ask for IMultipleResults because it returns 2 rowsets.
        hr = m_pINewOrderCommand[iHandleIndex]->Execute(
IID_IMultipleResults,
NULL,
&m_NewOrderExecuteParams[iHandleIndex],
NULL,
(IUnknown **)&pMultipleResults);
        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eExecute, "NewOrder()");
        }
        //////////////////////////////////////
        // Get order line results
        //////////////////////////////////////
        m_txn.NewOrder.total_amount = 0;
        for (i = 0; i < m_txn.NewOrder.o_ol_cnt; ++i)
        {
            // Get the first rowset object
            hr = pMultipleResults->GetResult(NULL, 0, IID_IRowset,
&IRowsAffected, (IUnknown **)&pRowset);
            if (FAILED(hr))
            {
                char szTmp[256];

                _snprintf(szTmp, sizeof(szTmp), "NewOrder() result set
%d, hr=0x%X", i, hr);

```

```

        ThrowError(m_pINewOrderCommand[m_txn.NewOrder.o_ol_cnt - 1],
COLEDBERR::eGetResult, szTmp);
    }

    // Fetch the result row handle(s)
    hr = pRowset->GetNextRows(DB_NULL_HCHAPTER, 0, cRows,
&cRowsObtained, &prghRows);
    if (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetNextRows, "NewOrder()");
    }

    // Fetch the actual row data by handle
    hr = pRowset->GetData(rghRows,
m_hNewOrderOutputAccessor[iHandleIndex], &m_txn.NewOrder.OL[i]);
    if (FAILED(hr))
    {
        ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetData, "NewOrder()");
    }

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

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

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

//////////
// Get the second rowset object
//////////
hr = pMultipleResults->GetResult(NULL, 0, IID_IRowset, &lRowsAffected,
(IUnknown **)&pRowset2);
if (FAILED(hr))
{
    char szTmp[256];

    _snprintf(szTmp, sizeof(szTmp), "NewOrder() result set %d,
hr=%d", i, hr);

    ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetResult, szTmp);
}

// Fetch the result row handle(s)
hr = pRowset2->GetNextRows(DB_NULL_HCHAPTER, 0, cRows2,
&cRowsObtained2, &prghRows2);

```

```

        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetNextRows, "NewOrder()");
        }

        // Fetch the actual row data by handle
        hr = pRowset2->GetData(rgRows2,
m_hNewOrderOutputAccessor2[iHandleIndex], &m_txn.NewOrder);
        if (FAILED(hr))
        {
            ThrowError(m_pINewOrderCommand[iHandleIndex],
COLEDBERR::eGetData, "NewOrder()");
        }

        // Release row(s)
        hr = pRowset2->ReleaseRows(cRowsObtained2, prghRows2, NULL, NULL,
NULL);

        // Release rowset
        hr = pRowset2->Release();

        // Release the common MultipleResults interface
        hr = pMultipleResults->Release();

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

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

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

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

```

```

void CTPCC_OLEDB::InitPaymentParams()
{
    int                i;
    HRESULT            hr;
    wchar_t            szName[iMAX_SP_NAME_LEN];
    IAccessor*         pIAccessor;
    const ULONG        nInputParams = 7;    // input parameters
    const ULONG        nOutputParams = 27;  // output result set
columns
    // Structure to bind in accessor
    DBBINDING          acInputDBBinding[nInputParams];
    DBBINDSTATUS        acInputDBBindStatus[nInputParams];
    DBBINDING          acOutputDBBinding[nOutputParams];
    DBBINDSTATUS        acOutputDBBindStatus[nOutputParams];

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

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

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

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

    // Payment parameter 2
    SetBinding(&acInputDBBinding[i++], offsetof(PAYMENT_DATA, c_w_id),
sizeof(m_txn.Payment.c_w_id), DBTYPE_I4);

    // Payment parameter 3
    SetBinding(&acInputDBBinding[i++], offsetof(PAYMENT_DATA, h_amount),
sizeof(m_txn.Payment.h_amount), DBTYPE_R8);

    // Payment parameter 4
    SetBinding(&acInputDBBinding[i++], offsetof(PAYMENT_DATA, d_id),
sizeof(m_txn.Payment.d_id), DBTYPE_UI1);

    // Payment parameter 5
    SetBinding(&acInputDBBinding[i++], offsetof(PAYMENT_DATA, c_d_id),
sizeof(m_txn.Payment.c_d_id), DBTYPE_UI1);

    // Payment parameter 6
    SetBinding(&acInputDBBinding[i++], offsetof(PAYMENT_DATA, c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

    // Payment parameter 7

```

```

        SetBinding(&acInputDBBinding[i++],          offsetof(PAYMENT_DATA,          c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

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

        hr = pIAccessor->CreateAccessor(
            DBACCESSOR_PARAMETERDATA,
            nInputParams,
            acInputDBBinding,
            sizeof(PAYMENT_DATA),
            &m_hPaymentInputAccessor,
            acInputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitPaymentParams()");
        }

        m_PaymentExecuteParams.cParamSets = 1;
        m_PaymentExecuteParams.hAccessor = m_hPaymentInputAccessor;
        m_PaymentExecuteParams.pData = &m_txn.Payment;

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

        i = 0;
        // Payment output column 1
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          c_id),
sizeof(m_txn.Payment.c_id), DBTYPE_I4);

        // Payment output column 2
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          c_last),
sizeof(m_txn.Payment.c_last), DBTYPE_STR);

        // Payment output column 3
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          h_date),
sizeof(m_txn.Payment.h_date), DBTYPE_DBTIMESTAMP);

        // Payment output column 4
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          w_street_1),
sizeof(m_txn.Payment.w_street_1), DBTYPE_STR);

        // Payment output column 5
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          w_street_2),
sizeof(m_txn.Payment.w_street_2), DBTYPE_STR);

        // Payment output column 6
        SetBinding(&acOutputDBBinding[i++],          offsetof(PAYMENT_DATA,          w_city),
sizeof(m_txn.Payment.w_city), DBTYPE_STR);

```

```

// Payment output column 7
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      w_state),
sizeof(m_txn.Payment.w_state), DBTYPE_STR);

// Payment output column 8
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      w_zip),
sizeof(m_txn.Payment.w_zip), DBTYPE_STR);

// Payment output column 9
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

// Payment output column 10
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 11
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 12
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

// Payment output column 13
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

// Payment output column 14
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_first),
sizeof(m_txn.Payment.c_first), DBTYPE_STR);

// Payment output column 15
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_middle),
sizeof(m_txn.Payment.c_middle), DBTYPE_STR);

// Payment output column 16
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_street_1),
sizeof(m_txn.Payment.d_street_1), DBTYPE_STR);

// Payment output column 17
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_street_2),
sizeof(m_txn.Payment.d_street_2), DBTYPE_STR);

// Payment output column 18
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_city),
sizeof(m_txn.Payment.d_city), DBTYPE_STR);

// Payment output column 19
SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_state),
sizeof(m_txn.Payment.d_state), DBTYPE_STR);

```



```

        // Payment output column 20
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      d_zip),
sizeof(m_txn.Payment.d_zip), DBTYPE_STR);

        // Payment output column 21
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_phone),
sizeof(m_txn.Payment.c_phone), DBTYPE_STR);

        // Payment output column 22
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_since),
sizeof(m_txn.Payment.c_since), DBTYPE_DBTIMESTAMP);

        // Payment output column 23
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_credit),
sizeof(m_txn.Payment.c_credit), DBTYPE_STR);

        // Payment output column 24
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_credit_lim),
sizeof(m_txn.Payment.c_credit_lim), DBTYPE_R8);

        // Payment output column 25
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_discount),
sizeof(m_txn.Payment.c_discount), DBTYPE_R8);

        // Payment output column 26
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_balance),
sizeof(m_txn.Payment.c_balance), DBTYPE_R8);

        // Payment output column 27
        SetBinding(&acOutputDBBinding[i++],      offsetof(PAYMENT_DATA,      c_data),
sizeof(m_txn.Payment.c_data), DBTYPE_STR);

        hr = pIAccessor->CreateAccessor(
            DBACCESSOR_ROWDATA | DBACCESSOR_OPTIMIZED,
            nOutputParams,
            acOutputDBBinding,
            sizeof(PAYMENT_DATA),
            &m_hPaymentOutputAccessor,
            acOutputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitPaymentParams()");
        }
    }

void CTPCC_OLEDB::Payment()
{
    HRESULT          hr;
    int              iTryCount = 0;
    IRowset*        pRowset;
    LONG             cRows = 1;    // number of rows returned in the rowset
    ULONG           cRowsObtained;
    HROW            rghRow;        //returned row handles

```

```

HROW*                prghRow = &rghRow;

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

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

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

        // Fetch the actual row data by handle
        hr = pRowset->GetData(rghRow, m_hPaymentOutputAccessor,
&m_txn.Payment);
        if (FAILED(hr))
        {
            ThrowError(m_pIPaymentCommand, COLEDBERR::eGetData,
"Payment()");
        }

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

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

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

        break;
    }
}

```

```

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

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

//      if (iTryCount)
//      throw
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

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

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

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

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

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

    // OrderStatus parameter 2

```

```

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

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

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

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

        hr = pIAccessor->CreateAccessor(
            DBACCESSOR_PARAMETERDATA,
            nInputParams,
            acInputDBBinding,
            sizeof(ORDER_STATUS_DATA),
            &m_hOrderStatusInputAccessor,
            acInputDBBindStatus);
        if (FAILED(hr))
        {
            ThrowError(pIAccessor,      COLEDBERR::eCreateAccessor,
"InitOrderStatusParams()");
        }

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

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

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

```

```

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

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

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

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

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

    while (TRUE)
    {
        try
        {
            // Execute the prepared command

```

```

        // Ask for IMultipleResults because it returns 2 rowsets.
        hr = m_pIOrderStatusCommand->Execute(NULL, IID_IMultipleResults,
&m_OrderStatusExecuteParams, NULL,

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

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

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

        m_txn.OrderStatus.o_ol_cnt = (short)cRowsObtained;

        // Get the data from multiple rows in this rowset
        for (i = 0; i < m_txn.OrderStatus.o_ol_cnt; ++i)
        {
            // Fetch the actual row data by handle
            hr = pRowset->GetData(rghRows[i],
m_hOrderStatusOutputAccessor, &m_txn.OrderStatus.OL[i]);
            if (FAILED(hr))
            {
                ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
            }
        }

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

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

```

```

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

    // Fetch the result row handle(s)
    hr = pRowset2->GetNextRows(DB_NULL_HCHAPTER, 0,
cRows2, &cRowsObtained2, &prghRows2);
    if (FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetNextRows, "OrderStatus()");
    }

    // Fetch the actual row data by handle
    hr = pRowset2->GetData(rghRows2,
m_hOrderStatusOutputAccessor2, &m_txn.OrderStatus);
    if (FAILED(hr))
    {
        ThrowError(m_pIOrderStatusCommand,
COLEDBERR::eGetData, "OrderStatus()");
    }

    // Release row(s)
    hr = pRowset2->Release();
}

// Release the common MultipleResults interface
hr = pMultipleResults->Release();

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

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

```



```

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

//      if (iTryCount)
//          throw
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

void CTPCC_OLEDB::InitDeliveryParams()
{
    int                i;
    HRESULT            hr;
    wchar_t            szName[iMAX_SP_NAME_LEN];
    IAccessor*         pIAccessor;
    const ULONG        nInputParams = 2;    // input parameters
    const ULONG        nOutputParams = 10; // output 1st result set
    columns
    // Structure to bind in accessor
    DBBINDING          acInputDBBinding[nInputParams];
    DBBINDSTATUS        acInputDBBindStatus[nInputParams];
    DBBINDING          acOutputDBBinding[nOutputParams];
    DBBINDSTATUS        acOutputDBBindStatus[nOutputParams];

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

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

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

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

    // Delivery parameter 2
    SetBinding(&acInputDBBinding[i++],    offsetof(DELIVERY_DATA,    o_carrier_id),
    sizeof(m_txn.Delivery.o_carrier_id), DBTYPE_I2);

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

```

```

}

hr = pIAccessor->CreateAccessor(
    DBACCESSOR_PARAMETERDATA,
    nInputParams,
    acInputDBBinding,
    sizeof(DELIVERY_DATA),
    &m_hDeliveryInputAccessor,
    acInputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
}

m_DeliveryExecuteParams.cParamSets = 1;
m_DeliveryExecuteParams.hAccessor = m_hDeliveryInputAccessor;
m_DeliveryExecuteParams.pData = &m_txn.Delivery;

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

// Binding for a rowset that may return more than one row.
for (i = 0; i < 10; ++i)
{
    // Delivery output column 1
    SetBinding(&acOutputDBBinding[i],    offsetof(DELIVERY_DATA,    o_id[i]),
sizeof(m_txn.Delivery.o_id[i]), DBTYPE_I4);
}

hr = pIAccessor->CreateAccessor(
    DBACCESSOR_ROWDATA | DBACCESSOR_OPTIMIZED,
    nOutputParams,
    acOutputDBBinding,
    sizeof(DELIVERY_DATA),
    &m_hDeliveryOutputAccessor,
    acOutputDBBindStatus);
if (FAILED(hr))
{
    ThrowError(pIAccessor, COLEDBERR::eCreateAccessor, "InitDeliveryParams()");
}
}

void CTPCC_OLEDB::Delivery()
{
    HRESULT          hr;
    int              iTryCount = 0;
    IRowset*        pRowset;
    LONG             cRows = 1;    // number of rows returned in the rowset
    ULONG           cRowsObtained;
    HROW            rghRow;        //returned row handles
    HROW*           prghRow = &rghRow;

    while (TRUE)

```

```

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

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

            // Fetch the actual row data by handle
            hr = pRowset->GetData(rghRow, m_hDeliveryOutputAccessor,
&m_txn.Delivery);
            if (FAILED(hr))
            {
                ThrowError(m_pIDeliveryCommand, COLEDBERR::eGetData,
"Delivery()");
            }

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

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

            m_txn.Delivery.exec_status_code = eOK;

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

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

```

```

//      if (iTryCount)
//          throw
CTPCC_OLEDB_ERR(CTPCC_OLEDB_ERR::ERR_RETRIED_TRANS, iTryCount);
}

```

new

tpcc_oledb.h

```

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

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

#define iMAX_SP_NAME_LEN 256 //maximum length of a stored procedure name with
parameters

// Type of parameter and result set column bindings.
enum eBindingType
{
    eInputParameter,
    eOutputParameter,
    eInputOutputParameter,
    eOutputColumn
};

class COLEDBERR : public CBaseErr
{
public:
    enum ACTION
    {
        eNone,
        eUnknown,
        eQueryInterface, // error from QueryInterface
        eCreateSession,
        eCreateCommand,
        eSetCommandText,
        eExecute, // = 6
        eCreateAccessor,
    }
};

```

```

        ePrepare,
        eGetNextRows,
        eGetData,
        eGetResult                // = 11
    };

    COLEDBERR(LPCTSTR szLoc)
        : CBaseErr(szLoc)
    {
        m_eAction = eNone;
        m_NativeError = 0;
        m_bDeadLock = FALSE;
        m_OLEDBErrStr = NULL;
    };

    ~COLEDBERR()
    {
        if (m_OLEDBErrStr != NULL)
            delete [] m_OLEDBErrStr;
    };

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

    int         ErrorType() {return ERR_TYPE_OLEDB;};
    char*       ErrorTypeStr() { return "OLEDB"; }
    int         ErrorNum() {return m_NativeError;};
    char*       ErrorText() {return m_OLEDBErrStr;};
    int         ErrorAction() { return (int)m_eAction; }

};

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

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

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

```

```

        int          m_errno;
        int          m_iTryCount;

        int          ErrorType() {return ERR_TYPE_TPCC_OLEDB;};
        char*       ErrorTypeStr() { return "TPCC OLEDB"; }
        int          ErrorNum() {return m_errno;};

        char*       ErrorText();
};

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

        DBPROPSET     m_rgInitPropSet;     //
initialization property set used to establish a connection
        DBPROP        m_InitProperties[4]; //
individual initialization properties

        IDBCreateSession* m_pIDBCreateSession; // session (connection)
interface
        IDBCreateCommand* m_pIDBCreateCommand; // SQL command
creation interface

        IMalloc*       m_pIMalloc;
        // Needed to release error strings.

        // StockLevel
        ICommandText* m_pIStockLevelCommand;
        HACCESSOR     m_hStockLevelInputAccessor; //
accessor to bind input parameters
        HACCESSOR     m_hStockLevelOutputAccessor; //
accessor to bind output columns
        DBPARAMS      m_StockLevelExecuteParams;
        // parameter structure for Execute

        // NewOrder
        // One prepared command for each possible number of new order line items
        ICommandText*
m_pINewOrderCommand[MAX_OL_NEW_ORDER_ITEMS];
        // accessors to bind input parameters
        // one for each possible number of new order line items
        HACCESSOR
m_hNewOrderInputAccessor[MAX_OL_NEW_ORDER_ITEMS];
        // accessor to bind output columns of the first rowset
        HACCESSOR
m_hNewOrderOutputAccessor[MAX_OL_NEW_ORDER_ITEMS];
        // accessor to bind output columns of the second rowset

```

```

        HACCESSOR
m_hNewOrderOutputAccessor2[MAX_OL_NEW_ORDER_ITEMS];
    // parameter structure for Execute
    DBPARAMS
m_NewOrderExecuteParams[MAX_OL_NEW_ORDER_ITEMS];

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

    // OrderStatus
    ICommandText*           m_pIOrderStatusCommand;
    HACCESSOR               m_hOrderStatusInputAccessor; //
accessor to bind input parameters
    HACCESSOR               m_hOrderStatusOutputAccessor; //
accessor to bind output columns
    HACCESSOR               m_hOrderStatusOutputAccessor2;
    // accessor to bind output columns
    DBPARAMS               m_OrderStatusExecuteParams;
    // parameter structure for Execute

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

    wchar_t                 m_szSPPrefix[32];           //      stored
procedures prefix

    // new-order specific fields
    int                     m_no_commit_flag;

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

    void CheckSPVersion();

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

    // Helper function to create and prepare a command

```

```

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

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

        // Helper function to initialize an array of bindings
        void InitBindings(DBBINDING* pDBBindings, int iCount, eBindingType
BindingType);

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

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

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

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

};

// wrapper routine for class constructor
extern "C" DllDecl CTPCC_OLEDB* CTPCC_OLEDB_new
( LPCSTR szServer, LPCSTR szUser, LPCSTR szPassword, LPCSTR szHost, LPCSTR
szDatabase, LPCWSTR szSPPrefix );

```



```
typedef CTPCC_OLEDB* (TYPE_CTPCC_OLEDB)(LPCSTR, LPCSTR, LPCSTR, LPCSTR,  
LPCSTR, LPCWSTR);
```

install.c

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

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

```
#include "resource.h"
```

```
#define WM_INITTEXT          WM_USER+100
```

```
HICON          hIcon;  
HINSTANCE      hInst;
```

```
DWORD          versionExeMS;  
DWORD          versionExeLS;  
DWORD          versionExeMM;  
DWORD          versionDIIMS;  
DWORD          versionDIILS;
```

```
// TPC-C registry settings  
TPCCREGISTRYDATA Reg;
```

```
static int      iPoolThreadLimit;  
static int      iMaxPoolThreads;  
static int      iThreadTimeout;  
static int      iListenBackLog;  
static int      iAcceptExOutstanding;
```

```

static int iUriEnableCache;
static int iUriScavengerPeriod;
static int iMaxConnections;

static int iIISMajorVersion;
static int iNumberOfProcessors;

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

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

SYSTEM_INFO siSysInfo;

BOOL install_com(char *szDllPath);

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

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

    hInst = hInstance;

    InitCommonControls();

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

    iRc = DialogBox(hInstance, MAKEINTRESOURCE(IDD_DIALOG4),
    GetDesktopWindow(), LicenseDlgProc);
    if ( iRc )

```

```

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

    DestroyIcon(hIcon);
    return 0;
}

```

```

BOOL CALLBACK LicenseDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)

```

```

{
    HGLOBAL          hRes;
    HRSRC            hResInfo;
    BYTE             *pSrc, *pDst;
    DWORD            dwSize;
    static HFONT     hFont;

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

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

```

```

        EndDialog(hwnd, TRUE);
    if ( wParam == IDCANCEL )
        EndDialog(hwnd, FALSE);
    default:
        break;
    }
    return FALSE;
}

```

```

BOOL CALLBACK UpdatedDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)

```

```

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

```

```

BOOL CALLBACK MainDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
lParam)

```

```

{
    PAINTSTRUCT        ps;
    MEMORYSTATUS        memoryStatus;
    OSVERSIONINFO        VI;
    char                szTmp[256];
    static char          szDllPath[256];
    static char          szWindowsPath[256];
    static char          szExePath[256];

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

            if ( GetWindowsInstallPath(szWindowsPath) )
            {

```

```

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

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

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

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

    ReadTPCCRegistrySettings( &Reg );
    ReadRegistrySettings();

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

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

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

    SetDlgItemText(hwnd, IDC_PATH, szDllPath);

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

```

```

FALSE);
        SetDlgItemInt(hwnd, ED_THREADS, Reg.dwNumberOfDeliveryThreads,
FALSE);
        SetDlgItemInt(hwnd, ED_MAXCONNECTION, Reg.dwMaxConnections,
FALSE);
        SetDlgItemInt(hwnd, ED_MAXDELIVERIES,
Reg.dwMaxPendingDeliveries, FALSE);
        SetDlgItemInt(hwnd, ED_IIS_MAX_THREAD_POOL_LIMIT,
iPoolThreadLimit, FALSE);
        SetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT, iThreadTimeout,
FALSE);
        SetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, iListenBackLog,
FALSE);
        SetDlgItemInt(hwnd, ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,
iAcceptExOutstanding, FALSE);

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

// check OS version level for COM. Must be at least Windows 2000
VI.dwOSVersionInfoSize = sizeof(VI);
GetVersionEx( &VI );
if (VI.dwMajorVersion < 5)
{
    HWND hDlg = GetDlgItem( hwnd, IDC_TM_MTS );
    EnableWindow( hDlg, 0 ); // disable COM option
    if (Reg.eTxnMon == COM)
        Reg.eTxnMon = None;
}

CheckDlgButton(hwnd, IDC_TM_NONE, 0);
CheckDlgButton(hwnd, IDC_TM_TUXEDO, 0);
CheckDlgButton(hwnd, IDC_TM_MTS, 0);
CheckDlgButton(hwnd, IDC_TM_ENCINA, 0);
switch (Reg.eTxnMon)
{
case None:
    CheckDlgButton(hwnd, IDC_TM_NONE, 1);
    break;
case TUXEDO:
    CheckDlgButton(hwnd, IDC_TM_TUXEDO, 1);
    break;
case ENCINA:
    CheckDlgButton(hwnd, IDC_TM_ENCINA, 1);
    break;
case COM:
    CheckDlgButton(hwnd, IDC_TM_MTS, 1);
    break;
}

```

```

        return TRUE;
    case WM_PAINT:
        if ( IsIconic(hwnd) )
        {
            BeginPaint(hwnd, &ps);
            DrawIcon(ps.hdc, 0, 0, hIcon);
            EndPaint(hwnd, &ps);
            return TRUE;
        }
        break;
    case WM_COMMAND:
        if ( HIWORD(wParam) == BN_CLICKED )
        {
            switch( LOWORD(wParam) )
            {
                case IDC_DBLIB:
                    return TRUE;
                case IDC_ODBC:
                    return TRUE;
                case IDOK:
                    ProcessOK(hwnd, szDllPath, szWindowsPath);
                    return TRUE;
                case IDCANCEL:
                    EndDialog(hwnd, FALSE);
                    return TRUE;
                default:
                    return FALSE;
            }
        }
        break;
    default:
        break;
    }
    return FALSE;
}

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

    char  szFullName[256];
    char  szErrTxt[128];

    // read settings from dialog
    Reg.dwNumberOfDeliveryThreads = GetDlgItemInt(hwnd, ED_THREADS, &d, FALSE);
    Reg.dwMaxConnections = GetDlgItemInt(hwnd, ED_MAXCONNECTION, &d, FALSE);
    Reg.dwMaxPendingDeliveries = GetDlgItemInt(hwnd, ED_MAXDELIVERIES, &d,
FALSE);

    GetDlgItemText(hwnd, ED_DB_SERVER, Reg.szDbServer, sizeof(Reg.szDbServer));

```

```

    GetDlgItemText(hwnd, ED_DB_USER_ID, Reg.szDbUser, sizeof(Reg.szDbUser));
    GetDlgItemText(hwnd, ED_DB_PASSWORD, Reg.szDbPassword,
sizeof(Reg.szDbPassword));
    GetDlgItemText(hwnd, ED_DB_NAME, Reg.szDbName, sizeof(Reg.szDbName));

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

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

    iPoolThreadLimit = GetDlgItemInt(hwnd, ED_IIS_MAX_THREAD_POOL_LIMIT, &d,
FALSE);
    iThreadTimeout = GetDlgItemInt(hwnd, ED_IIS_THREAD_TIMEOUT, &d, FALSE);
    iListenBackLog = GetDlgItemInt(hwnd, ED_IIS_LISTEN_BACKLOG, &d, FALSE);
    iAcceptExOutstanding = GetDlgItemInt(hwnd,
ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE, &d, FALSE);

    ShowWindow(hwnd, SW_HIDE);
    hDlg = CreateDialog(hInst, MAKEINTRESOURCE(IDD_DIALOG3), hwnd, CopyDlgProc);
    ShowWindow(hDlg, SW_SHOWNA);
    UpdateDialog(hDlg);

    // check to see if the web services are running
    bSvcRunning = CheckWWWebService();
    if ( bSvcRunning )
    {
        SetDlgItemText(hDlg, IDC_STATUS, "Stopping Web Service.");
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

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

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

```



```

if ( !rc )
{
    ShowWindow(hwnd, SW_SHOWNA);
    DestroyWindow(hDlg);
    strcpy( szErrTxt, "Error(s) occured when creating " );
    strcat( szErrTxt, szLastFileName );
    MessageBox(hwnd, szErrTxt, NULL, MB_ICONSTOP | MB_OK);
    EndDialog(hwnd, 0);
    return;
}

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

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

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

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

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

```

```

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

    Sleep(100);

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

    EndDialog(hwnd, rc);
    return;
}

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

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

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, "SYSTEM\\CurrentControlSet\\Services\\Inetinfo\\Parameters", 0, KEY_READ, &hKey) ==
ERROR_SUCCESS )
    {
        if ( iIISMajorVersion == 6)
        {
            // since IIS6 handles the pool thread parameters differently, we need to fill in
the dialog
            // with the MaxPoolThreads rather than PoolThreadLimit
            // for ease of coding, we are just going to stuff the value into
iPoolThreadLimit
            size = sizeof(iPoolThreadLimit);
            if ( RegQueryValueEx(hKey, "MaxPoolThreads", 0, &type, (char
*)&iPoolThreadLimit, &size) == ERROR_SUCCESS )
                if ( !iPoolThreadLimit )
                    iPoolThreadLimit = iMaxPhysicalMemory * 2;
        }
    }
}

```

```

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

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

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

    RegCloseKey(hKey);
}

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

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

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

```

```

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

        RegCloseKey(hKey);
    }
}

```

```

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

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

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

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

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

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

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

```

```

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

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

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }

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

        RegFlushKey(hKey);
        RegCloseKey(hKey);
    }
}

```

```

        return;
    }

    BOOL CALLBACK CopyDlgProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM
    lParam)
    {
        if ( uMsg == WM_INITDIALOG )
        {
            SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETRANGE, 0,
            MAKELPARAM(0, 16));
            SendDlgItemMessage(hwnd, IDC_PROGRESS1, PBM_SETSTEP, (WPARAM)1, 0);
            return TRUE;
        }
        return FALSE;
    }

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

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

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

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

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

        dwSize = SizeofResource(hInst, hResInfo);
        hDLL = LoadResource(hInst, hResInfo );
    }

```

```

    pSrc = (BYTE *)LockResource(hDLL);
    remove(szFullName);

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

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

    CloseHandle(hFile);

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

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

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

    // install MSVCR70.DLL
    strcpy( szLastFileName, "msvcr70.dll" );
    if (!FileFromResource( "MSVCRT70", IDR_MSVCRT701, szWindowsPath,
szLastFileName ))
        return 0;
    SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
    UpdateDialog(hDlg);

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

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

```

```

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

        // install tpcc_tuxedo.dll
        strcpy( szLastFileName, "tpcc_tuxedo.dll" );
        if ( !FileFromResource( "TUXEDO_DLL", IDR_TUXEDO_DLL, szDllPath,
szLastFileName ) )
            return 0;
        //SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        //UpdateDialog(hDlg);

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

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

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

        // install tpcc_com_all.dll
        strcpy( szLastFileName, "tpcc_com_all.dll" );
        if ( !FileFromResource( "COM_ALL_DLL", IDR_COMALL_DLL, szDllPath,
szLastFileName ) )
            return 0;
        SendDlgItemMessage(hDlg, IDC_PROGRESS1, PBM_STEPIT, 0, 0);
        UpdateDialog(hDlg);

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

        return 1;
    }

static BOOL GetInstallPath(char *szDllPath)

```



```

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

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

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

        RegCloseKey(hKey);
    }

    return bRc;
}

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

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

```

```

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

        RegCloseKey(hKey);
    }

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

    versionDllMS = 0;
    versionDllLS = 0;
    if ( _access(szDLLPath, 00) == 0 )
    {
        dwSize = GetFileVersionInfoSize(szDLLPath, &d);
        if ( dwSize )
        {
            ptr = (char *)malloc(dwSize);
            GetFileVersionInfo(szDLLPath, 0, dwSize, ptr);
            VerQueryValue(ptr, "\\",&vs, &dwBytes);
            versionDllMS = vs->dwProductVersionMS;
            versionDllLS = vs->dwProductVersionLS;
            free(ptr);
        }
    }

    versionExeMS = 0x7FFF;
    versionExeLS = 0x7FFF;
    dwSize = GetFileVersionInfoSize(szExePath, &d);

```

```

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

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

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

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

```

```

        return FALSE;

    if (! StartService(schService, 0, NULL) )
        goto StartWWWebErr;
    //start Service pending, Check the status until the service is running.
    if (! QueryServiceStatus(schService, &ssStatus) )
        goto StartWWWebErr;
    while( ssStatus.dwCurrentState != SERVICE_RUNNING)
    {
        dwOldCheckPoint = ssStatus.dwCheckPoint;           //Save the
current checkpoint.
        Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
        if ( !QueryServiceStatus(schService, &ssStatus) ) //Check the status again.
            break;
        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)     //Break if the checkpoint
has not been incremented.
            break;
    }

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

    CloseServiceHandle(schService);
    return TRUE;

StartWWWebErr:
    CloseServiceHandle(schService);
    return FALSE;
}

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

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

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

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

```

```

        goto StopWWWebErr;
while( ssStatus.dwCurrentState == SERVICE_RUNNING)
{
        dwOldCheckPoint = ssStatus.dwCheckPoint;           //Save the
current checkpoint.
        Sleep(ssStatus.dwWaitHint);
        //Wait for the specified interval.
        if ( !QueryServiceStatus(schService, &ssStatus) ) //Check the status again.
                break;
        if (dwOldCheckPoint >= ssStatus.dwCheckPoint)       //Break if the checkpoint
has not been incremented.
                break;
}

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

        CloseServiceHandle(schService);
        return TRUE;

StopWWWebErr:
        CloseServiceHandle(schService);
        return FALSE;
}

static void UpdateDialog(HWND hDlg)
{
        MSG msg;

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

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

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

        irc = system("IIS6_CONFIG.CMD");

        // since the return code from the command file is always 1,

```

```

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

```

install.h

```

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

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

#define BN_LOG                      1001
#define ED_KEEP                     1002
#define ED_THREADS                  1003
#define ED_THREADS2                 1004
#define IDC_PATH                    1007
#define IDC_VERSION                 1009
#define IDC_RESULTS                 1010
#define IDC_PROGRESS1              1011
#define IDC_STATUS                  1012
#define IDC_BUTTON1                 1013
#define ED_MAXCONNECTION            1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT       1018
#define ED_IIS_LISTEN_BACKLOG       1019
#define IDC_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 Visual C++ 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_SETFONT | DS_MODALFRAME | DS_CENTER | WS_MINIMIZEBOX | WS_POPUP
|
  WS_CAPTION | WS_SYSMENU
CAPTION "TPC-C Web Client Installation Utility"
FONT 8, "MS Sans Serif", 0, 0, 0x1
BEGIN
  EDITTEXT    ED_THREADS,164,45,34,12,ES_RIGHT | ES_NUMBER,
              WS_EXRTLREADING
  EDITTEXT    ED_MAXDELIVERIES,164,59,34,12,ES_RIGHT | ES_NUMBER,
              WS_EXRTLREADING
  EDITTEXT    ED_MAXCONNECTION,164,73,34,12,ES_RIGHT | ES_NUMBER,
              WS_EXRTLREADING
  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_THREAD_POOL_LIMIT,164,263,34,12,ES_RIGHT |
            ES_NUMBER,WS_EX_RTLCREADING
EDITTEXT    ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE,164,277,34,12,ES_RIGHT |
            ES_NUMBER,WS_EX_RTLCREADING
EDITTEXT    ED_IIS_THREAD_TIMEOUT,164,291,34,12,ES_RIGHT | ES_NUMBER,
            WS_EX_RTLCREADING
EDITTEXT    ED_IIS_LISTEN_BACKLOG,164,305,34,12,ES_RIGHT | ES_NUMBER,
            WS_EX_RTLCREADING
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_SETFONT | DS_SETFOREGROUND | DS_3DLOOK | DS_CENTER | WS_POPUP |
        WS_BORDER
EXSTYLE WS_EX_STATICEDGE
FONT 12, "MS Sans Serif", 0, 0, 0x1
BEGIN
    DEFPUSHBUTTON "OK",IDOK,33,45,50,9
    CTEXT         "HTML TPC-C Installation Successful",IDC_RESULTS,7,22,
                102,18,0,WS_EX_CLIENTEDGE
    ICON          IDI_ICON2,IDC_STATIC,50,7,18,20,SS_REALSIZEIMAGE,
                WS_EX_TRANSPARENT
END

```



```

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

```

```

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

```

```

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

```

```

#ifdef APSTUDIO_INVOKED
GUIDELINES DESIGNINFO
BEGIN
  IDD_DIALOG1, DIALOG
  BEGIN
    LEFTMARGIN, 22
    RIGHTMARGIN, 209
    VERTGUIDE, 35
    VERTGUIDE, 198
    TOPMARGIN, 4
    BOTTOMMARGIN, 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
BEGIN
    "resource.h\0"
END

```

```

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

```

```

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

```

```

#endif // APSTUDIO_INVOKED

```

```

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

```

```

// Icon with lowest ID value placed first to ensure application icon
// remains consistent on all systems.

```

```

IDI_ICON1        ICON        "icon1.ico"
IDI_ICON2        ICON        "icon2.ico"

```

```
////////////////////////////////////  
//  
// TPCCDLL  
//
```

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

```
////////////////////////////////////  
//  
// Version  
//
```

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

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

```
IDR_LICENSE1     LICENSE     "license.txt"
```

```

////////////////////////////////////
//
// DBLIB_DLL
//
IDR_DBLIB_DLL      DBLIB_DLL      "..\..\db_dblib_dll\bin\Release\tpcc_dblib.dll"

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

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

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

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

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

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

////////////////////////////////////
//
// COM_TYPLIB

```

```

//

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

////////////////////////////////////
//
// MSVCRT70
//

IDR_MSVCRT701       MSVCRT70          "C:\\WINDOWS\\system32\\msvcr70.dll"
#endif // English (U.S.) resources
////////////////////////////////////

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

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

```

install_com.cpp

```

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

#define _WIN32_WINNT 0x0500

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

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

```

```

}

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

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

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

    CoInitializeEx(NULL, COINIT_MULTITHREADED);

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

    if (!SUCCEEDED(hr)) goto Error;

    bstrTemp = "Applications";

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

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

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

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

```

```

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

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

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

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

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

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

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

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

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

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

```

```

if (!SUCCEEDED(hr)) goto Error;

pCatalogObjectApp->Release();
pCatalogObjectApp = NULL;

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

hr = pCOMAdminCat->InstallComponent(bstrTemp,

bstrTemp2,

bstrTemp3,

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

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

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

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

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

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

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

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

```



```

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

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

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

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

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

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

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

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

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

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

    hr = pCatalogCollectionMethod->Populate();

```

```

        if (!SUCCEEDED(hr)) goto Error;

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

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

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

            pCatalogObjectMethod->Release();
            pCatalogObjectMethod = NULL;

            lCountMethod--;
        }

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

        pCatalogObjectItf->Release();
        pCatalogObjectItf = NULL;

        lCountItf--;
    }

    pCatalogObjectCo->Release();
    pCatalogObjectCo = NULL;

    lCountCo--;
}

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

pCatalogCollectionApp->Release();
pCatalogCollectionApp = NULL;

pCatalogCollectionCo->Release();
pCatalogCollectionCo = NULL;

```

```

    pCatalogCollectionItf->Release();
    pCatalogCollectionItf = NULL;

    pCatalogCollectionMethod->Release();
    pCatalogCollectionMethod = NULL;

Error:
    CoUninitialize();

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

```

RESOURCE.H

```

//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by install.rc
//
#define IDD_DIALOG1          101
#define IDI_ICON1           102
#define IDR_TPCCDLL         103
#define IDD_DIALOG2        105
#define IDI_ICON2           106
#define IDR_DELIVERY        107
#define IDD_DIALOG3        108
#define IDR_LICENSE1       112
#define IDD_DIALOG4        113
#define IDR_TPCCOBJ1       117
#define IDR_TPCCSTUB1      118
#define IDR_DBLIB_DLL      122
#define IDR_ODBC_DLL       123
#define IDR_TUXEDO_APP     124
#define IDR_TUXEDO_DLL     125
#define IDR_COM_DLL        126
#define IDR_COMPS_DLL      127
#define IDR_COMALL_DLL     128

```

```

#define IDR_COMTYPLIB_DLL          129
#define IDR_MSVCRT701              130
#define BN_LOG                      1001
#define ED_KEEP                     1002
#define ED_THREADS                  1003
#define ED_THREADS2                 1004
#define IDC_PATH                    1007
#define IDC_VERSION                 1009
#define IDC_RESULTS                 1010
#define IDC_PROGRESS1               1011
#define IDC_STATUS                  1012
#define IDC_BUTTON1                 1013
#define ED_MAXCONNECTION            1014
#define ED_IIS_MAX_THREAD_POOL_LIMIT 1015
#define ED_MAXDELIVERIES            1016
#define ED_WEB_SERVICE_BACKLOG_QUEUE_SIZE 1017
#define ED_IIS_THREAD_TIMEOUT       1018
#define ED_IIS_LISTEN_BACKLOG       1019
#define IDC_DBLIB                   1021
#define IDC_LICENSE                  1022
#define IDC_ODBC                     1022
#define IDC_CONNECT_POOL            1023
#define ED_DB_SERVER                 1023
#define ED_USER_CONNECT_DELAY_TIME  1024
#define ED_DB_USER_ID                1024
#define IDC_MTS                      1025
#define IDC_TM_MTS                   1025
#define IDC_TM_TUXEDO                1026
#define IDC_TM_NONE                  1027
#define ED_DB_PASSWORD               1028
#define ED_DB_NAME                   1029
#define IDC_TM_ENCINA                1030

```

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

```
//
```

```

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

```

resource.h

```

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

```

```

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

```

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 "420"

static CRITICAL_SECTION TermCriticalSection;

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

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

// For deferred Delivery txns:

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

```

```

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

// configuration settings from registry
TPCCREGISTRYDATA     Reg;

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

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

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

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

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

```

```

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

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

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

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

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

                TermInit();

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

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

```



```

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

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

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

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

```

```

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

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

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

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

dwDelBuffFreeCount = dwDelBuffSize;

InitJulianTime(NULL);

// create unique log file name based on delilog-yymmdd-
hhmm.log
SYSTEMTIME Time;
GetLocalTime( &Time );
wsprintf( szLogFile, "%sdelivery-%2.2d%2.2d%2.2d-
%2.2d%2.2d-%2.2ds%2.2dms.log",
Reg.szPath, Time.wYear % 100, Time.wMonth,
Time.wDay, Time.wHour, Time.wMinute, Time.wSecond, Time.wMilliseconds );
txnDelilog = new CTxnLog(szLogFile,
TXN_LOG_WRITE);

//write event into txn log for START
txnDelilog->WriteCtrlRecToLog(TXN_EVENT_START,
szMyComputerName, sizeof(szMyComputerName));

// allocate structures for delivery buffers and thread mgmt
pDeliHandles = new HANDLE[dwNumDeliveryThreads];
pDelBuff = new
DELIVERY_TRANSACTION[dwDelBuffSize];
// launch DeliveryWorkerThread to perform actual delivery
txns
for(i=0; i<dwNumDeliveryThreads; i++)
{

```

```

        pDeliHandles[i] = (HANDLE)
        _beginthread( DeliveryWorkerThread, 0, NULL );
        if (pDeliHandles[i] ==
INVALID_HANDLE_VALUE)
            throw new
CWEBCLNT_ERR( ERR_DELIVERY_THREAD_FAILED );
    }
}
break;

case DLL_PROCESS_DETACH:
    if (dwNumDeliveryThreads)
    {
        if (txnDelilog != NULL)
        {
            //write event into txn log for STOP
            txnDelilog->
>WriteCtrlRecToLog(TXN_EVENT_STOP, szMyComputerName, sizeof(szMyComputerName));

            // This will do a clean shutdown of the delivery log
            file
            CTxnLog *txnDelilogLocal = txnDelilog;
            txnDelilog= NULL;
            delete txnDelilogLocal;
        }

        delete [] pDeliHandles;
        delete [] pDelBuff;

        CloseHandle( hWorkerSemaphore );
        CloseHandle( hDoneEvent );
        DeleteCriticalSection(&DelBuffCriticalSection);

        // Delete delivery delay critical section
        //
        DeleteCriticalSection(&hConnectCriticalSection);
    }

    DeleteCriticalSection(&TermCriticalSection);

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

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

    Sleep(500);
    break;

default:

```

```

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

    _sntprintf(szMsg, sizeof(szMsg), "%s error, code %d: %s",
               e->ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
    WriteMessageToEventLog( szMsg );
    delete e;
    TerminateExtension(0);
    return FALSE;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled exception. DLL could not load.));
    TerminateExtension(0);
    return FALSE;
}

return TRUE;
}

```

/* FUNCTION: GetExtensionVersion

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

```

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

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

    return TRUE;
}

```

/* FUNCTION: TerminateExtension

*
* PURPOSE: This function is called by the inet service when the DLL is about to be unloaded.
* Release all resources in anticipation of being unloaded.
*

```

* RETURNS:          TRUE  inet service expected return value.
*/

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

    TermDeleteAll();
    return TRUE;
}

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

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

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

#ifdef ICECAP
    StartCAP();
#endif

    try
    {

```

```

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

if (TermId != 0)
{
    if ( TermId < 0 || TermId >= Term.iNumEntries ||
Term.pClientData[TermId].iNextFree != -1 )
    {
        // debugging...
        char szTmp[128];
        wsprintf( szTmp, "Invalid term ID; TermId = %d", TermId );
        WriteMessageToEventLog( szTmp );

        throw new CWEBCLNT_ERR( ERR_INVALID_TERMID );
    }

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

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

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

```

```

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

```

```

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

#ifdef ICECAP
    StopCAP();
#endif

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

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

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

```

```

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

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

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

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

```



```

    (VOID) DeregisterEventSource(hEventSource);
}
}

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

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

    DELIVERY_TRANSACTION      delivery;
    PDELIVERY_DATA            pDeliveryData;
    TXN_RECORD_TPCC_DELIV_DEF  txnDeliRec;

    DWORD                  index;
    HANDLE                  handles[2];

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

    assert(txnDeliRec != NULL);

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

                Sleep(Reg.dwConnectDelay);
            }

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

            if (Reg.dwConnectDelay > 0)
            {

```

```

        // Synchronize connect (for VIA)
        //
        LeaveCriticalSection(&hConnectCriticalSection);
    }
}
else
{
    if (Reg.eDB_Protocol == DBLIB)
        pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );
    }
    pDeliveryData = pTxn->BuffAddr_Delivery();
}
catch (CBaseErr *e)
{
    char szTmp[1024];
    wsprintf( szTmp, "Error in Delivery Txn thread. Could not connect to database. "
"%s. Server=%s, User=%s, Password=%s, Database=%s",
e->ErrorText(), Reg.szDbServer, Reg.szDbUser, Reg.szDbPassword,
Reg.szDbName );
    WriteMessageToEventLog( szTmp );
    delete e;
    goto ErrorExit;
}
catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread.));
    goto ErrorExit;
}

while (TRUE)
{
    try
    {
        //while delivery thread running, i.e. user has not requested termination
        while (TRUE)
        {
            // need to wait for multiple objects: program exit or worker
semaphore;

            handles[0] = hDoneEvent;
            handles[1] = hWorkerSemaphore;
            index = WaitForMultipleObjects( 2, &handles[0], FALSE,
INFINITE );

            if (index == WAIT_OBJECT_0)
                goto ErrorExit;

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

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

```

```

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

        LeaveCriticalSection(&DelBuffCriticalSection);

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

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

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

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

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

    // log the error txn
    txnDeliRec.TxnStatus = e->ErrorType();
    if (txnDelilog != NULL)
        txnDelilog->WriteToLog(&txnDeliRec);

    delete e;
}
catch (...)
{
    // unhandled exception; shouldn't happen; not much we can do...
    WriteMessageToEventLog(TEXT("Unhandled exception caught in
DeliveryWorkerThread."));
}

```

```

    }
ErrorExit:
    if (Reg.dwConnectDelay > 0)
    {
        // Synchronize disconnect (for VIA)
        //
        EnterCriticalSection(&hConnectCriticalSection);

        Sleep(Reg.dwConnectDelay);
    }

    delete pTxn;

    if (Reg.dwConnectDelay > 0)
    {
        // Synchronize disconnect (for VIA)
        //
        LeaveCriticalSection(&hConnectCriticalSection);
    }

    _endthread();
}

/* FUNCTION: PostDeliveryInfo
 *
 * PURPOSE:   This function enters the delivery txn into the deferred delivery buffer.
 *
 * RETURNS:   BOOL  FALSE  delivery information posted successfully
 *            TRUE   error cannot post delivery info
 */
BOOL PostDeliveryInfo(long w_id, short o_carrier_id)
{
    BOOL bError;

    EnterCriticalSection(&DelBuffCriticalSection);
    if (dwDelBuffFreeCount > 0)
    {
        bError = FALSE;
        (pDelBuff+dwDelBuffFreeIndex)->w_id          = w_id;
        (pDelBuff+dwDelBuffFreeIndex)->o_carrier_id  = 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 *szCmnds[] =
    {
        "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=\"TERMIN\"
VALUE=\"0\">"
                                "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\"
VALUE=\"0\">"
                                "<INPUT                                TYPE=\"hidden\"
NAME=\"VERSION\" VALUE=\"\" WEBCLIENT_VERSION \">"
                                );
    sprintf( szTmp, "Configuration    Settings:  <BR><font    face=\"Courier    New\"
color=\"blue\"><PRE>"
            "Txn Monitor            = <B>%s</B><BR>"
            "Database protocol        = <B>%s</B><BR>"
            "Max Connections            = <B>%d</B><BR>"
            "# of Delivery Threads      = <B>%d</B><BR>"

```

```

                "Max Pending Deliveries = <B>%d</B><BR>"
                , szTxnMonNames[Reg.eTxnMon], szDBNames[Reg.eDB_Protocol],
                Reg.dwMaxConnections, dwNumDeliveryThreads, dwDelBuffSize );
strcat( szBuffer, szTmp);

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

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

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

```

```

VALUE="Submit">"
                                                                    "</FORM></BODY></HTML>");
}

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

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

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

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

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

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

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

```



```

iNewTerm = TermAdd();

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

try
{
    if (Reg.eTxnMon == TUXEDO)
        Term.pClientData[iNewTerm].pTxn = pCTPCC_TUXEDO_new();
    else if (Reg.eTxnMon == ENCINA)
        Term.pClientData[iNewTerm].pTxn = pCTPCC_ENCINA_new();
    else if (Reg.eTxnMon == COM)
        Term.pClientData[iNewTerm].pTxn =
pCTPCC_COM_new( Reg.bCOM_SinglePool );
    else if (Reg.eDB_Protocol == ODBC)
        Term.pClientData[iNewTerm].pTxn = pCTPCC_ODBC_new(
szServer,
szUser, szPassword, szMyComputerName,

szDatabase, Reg.szSPPrefix,

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

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

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

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

    EnterCriticalSection(&TermCriticalSection);

    iTot = 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,
          "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,
          "Carrier ID invalid must be numeric 1 - 10."
        },
        {      ERR_DELIVERY_MISSING_OCD_KEY,
          "Delivery missing Carrier ID key \"OCD*\"."
        },
        {      ERR_DELIVERY_THREAD_FAILED,
          "not start delivery worker thread."
        },
        {      ERR_GETPROCADDR_FAILED,
          "Could not map proc in DLL. GetProcAddress error. DLL="
        },
        {      ERR_HTML_ILL_FORMED,
          "Required key field is missing from HTML string."
        },
        {      ERR_INVALID_SYNC_CONNECTION,
          "Terminal Sync ID."
        },
        {      ERR_INVALID_TERMID,
          "Invalid Terminal ID."
        },
        {      ERR_LOADDLL_FAILED,
          "Load of DLL failed. DLL="
        },
        {      ERR_MAX_CONNECTIONS_EXCEEDED,
          "connections available. Max Connections is probably too low."
        },
        {      ERR_MISSING_REGISTRY_ENTRIES,
          "registry entries are missing. Rerun INSTALL to correct."
        },
        {      ERR_NEWORDER_CUSTOMER_INVALID,
          "Order customer id invalid data type, range = 1 to 3000."
        },
        {      ERR_NEWORDER_CUSTOMER_KEY,
          "Order missing Customer key \"CID*\"."
        },
        {      ERR_NEWORDER_DISTRICT_INVALID,
          "Order District ID Invalid range 1 - 10."
        }
    },
    "Invalid",
    "Delivery",
    "Could",
    "Invalid",
    "No",
    "Required",
    "New",
    "New",
    "New"
}

```

Order missing District key \"DID*\".	{	ERR_NEWORDER_FORM_MISSING_DID,	"New
	}		},
Item Id is wrong data type, must be numeric."	{	ERR_NEWORDER_ITEMID_INVALID,	"New
	}		Order
Order Item Id is out of range. Range = 1 to 999999."	{	ERR_NEWORDER_ITEMID_RANGE,	"New
	}		},
Order Item_Id field entered without a corresponding Supp_W."	{	ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,	"New
	}		},
Order missing Item Id key \"IID*\".	{	ERR_NEWORDER_MISSING_IID_KEY,	"New
	}		},
Order Missing Qty key \"Qty##*\".	{	ERR_NEWORDER_MISSING_QTY_KEY,	"New
	}		},
Order missing Supp_W key \"SP##*\".	{	ERR_NEWORDER_MISSING_SUPPW_KEY,	"New
	}		},
Order No order lines entered."	{	ERR_NEWORDER_NOITEMS_ENTERED,	"New
	}		},
Order Qty invalid must be numeric range 1 - 99."	{	ERR_NEWORDER_QTY_INVALID,	"New
	}		},
"New Order Qty is out of range. Range = 1 to 99."	{	ERR_NEWORDER_QTY_RANGE,	
	}		},
Order Qty field entered without a corresponding Supp_W."	{	ERR_NEWORDER_QTY_WITHOUT_SUPPW,	"New
	}		},
Order Supp_W invalid data type must be numeric."	{	ERR_NEWORDER_SUPPW_INVALID,	"New
	}		},
Server name specified."	{	ERR_NO_SERVER_SPECIFIED,	"No
	}		},
Only Customer ID or Last Name may be entered, not both."}	{	ERR_ORDERSTATUS_CID_AND_CLT,	"Order
	}		Status
Customer ID invalid, range must be numeric 1 - 3000."}	{	ERR_ORDERSTATUS_CID_INVALID,	"Order
	}		Status
Status Customer last name longer than 16 characters."}	{	ERR_ORDERSTATUS_CLT_RANGE,	"Order
	}		
District invalid, value must be numeric 1 - 10."}	{	ERR_ORDERSTATUS_DID_INVALID,	"Order
	}		Status
Either Customer ID or Last Name must be entered."}	{	ERR_ORDERSTATUS_MISSING_CID_CLT,	"Order
	}		Status
missing Customer key \"CID*\".	{	ERR_ORDERSTATUS_MISSING_CID_KEY,	"Order
	}		Status
missing Customer Last Name key \"CLT*\".	{	ERR_ORDERSTATUS_MISSING_CLT_KEY,	"Order
	}		Status
missing District key \"DID*\".	{	ERR_ORDERSTATUS_MISSING_DID_KEY,	"Order
	}		Status
"Payment Customer district invalid must be numeric."}	{	ERR_PAYMENT_CDI_INVALID,	
	}		},
"Payment Only Customer ID or Last Name may be entered, not both."}	{	ERR_PAYMENT_CID_AND_CLT,	
	}		},
"Payment Customer data type invalid, must be numeric."}	{	ERR_PAYMENT_CUSTOMER_INVALID,	
	}		},
"Payment Customer Warehouse invalid, must be numeric."}	{	ERR_PAYMENT_CWI_INVALID,	
	}		},

```

        {      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=\\\"TERMINID\\\" VALUE=\\\"%d\\\">"
    "<INPUT TYPE=\\\"hidden\\\" NAME=\\\"SYNCD\\\" VALUE=\\\"%d\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..NewOrder..\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..Payment..\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..Delivery..\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..Order-Status..\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..Stock-Level..\\\">"
    "<INPUT TYPE=\\\"submit\\\" NAME=\\\"CMD\\\" VALUE=\\\"..Exit..\\\">"
    "</FORM></BODY></HTML>"
    , MAIN_MENU_FORM, iTermId, iSyncId);
}

```

```

/* FUNCTION: MakeStockLevelForm

```

```

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

```

```

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

```

```

{
    int c;

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

    if ( bInput )
    {
        strcpy(szForm+c,
            "Stock Level Threshold: <INPUT NAME=\\\"TT*\\\" SIZE=2><BR> <BR>"
            "low stock: </font><BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
            <BR> <BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR></PRE><HR>"

```

```

        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
        "</FORM></HTML>");
    }
    else
    {
        wsprintf(szForm+c,
            "Stock Level Threshold: %2.2d<BR> <BR>"
            "low stock: %3.3d</font> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
            "<BR> <BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR></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=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"

```

```

" <PRE><font face=\"Courier\">
, bValid ? 0 : ERR_BAD_ITEM_ID, NEW_ORDER_FORM, iTermId,
Term.pClientData[iTermId].iSyncId);

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

strcpy( szForm+c,
"District: <INPUT NAME=\"DID*\" SIZE=1> Date:<BR>"
"Customer: <INPUT NAME=\"CID*\" SIZE=4> Name: Credit:
%Disc:<BR>"
"Order Number: Number of Lines: W_tax: D_tax:<BR>"
<BR>"
" Supp_W Item_Id Item Name Qty Stock B/G Price
Amount<BR>"
" <INPUT NAME=\"SP00*\" SIZE=4> <INPUT NAME=\"IID00*\"
SIZE=6> <INPUT NAME=\"Qty00*\" SIZE=1><BR>"
" <INPUT NAME=\"SP01*\" SIZE=4> <INPUT NAME=\"IID01*\"
SIZE=6> <INPUT NAME=\"Qty01*\" SIZE=1><BR>"
" <INPUT NAME=\"SP02*\" SIZE=4> <INPUT NAME=\"IID02*\"
SIZE=6> <INPUT NAME=\"Qty02*\" SIZE=1><BR>"
" <INPUT NAME=\"SP03*\" SIZE=4> <INPUT NAME=\"IID03*\"
SIZE=6> <INPUT NAME=\"Qty03*\" SIZE=1><BR>"
" <INPUT NAME=\"SP04*\" SIZE=4> <INPUT NAME=\"IID04*\"
SIZE=6> <INPUT NAME=\"Qty04*\" SIZE=1><BR>"
" <INPUT NAME=\"SP05*\" SIZE=4> <INPUT NAME=\"IID05*\"
SIZE=6> <INPUT NAME=\"Qty05*\" SIZE=1><BR>"
" <INPUT NAME=\"SP06*\" SIZE=4> <INPUT NAME=\"IID06*\"
SIZE=6> <INPUT NAME=\"Qty06*\" SIZE=1><BR>"
" <INPUT NAME=\"SP07*\" SIZE=4> <INPUT NAME=\"IID07*\"
SIZE=6> <INPUT NAME=\"Qty07*\" SIZE=1><BR>"
" <INPUT NAME=\"SP08*\" SIZE=4> <INPUT NAME=\"IID08*\"
SIZE=6> <INPUT NAME=\"Qty08*\" SIZE=1><BR>"
" <INPUT NAME=\"SP09*\" SIZE=4> <INPUT NAME=\"IID09*\"
SIZE=6> <INPUT NAME=\"Qty09*\" SIZE=1><BR>"
" <INPUT NAME=\"SP10*\" SIZE=4> <INPUT NAME=\"IID10*\"
SIZE=6> <INPUT NAME=\"Qty10*\" SIZE=1><BR>"
" <INPUT NAME=\"SP11*\" SIZE=4> <INPUT NAME=\"IID11*\"
SIZE=6> <INPUT NAME=\"Qty11*\" SIZE=1><BR>"
" <INPUT NAME=\"SP12*\" SIZE=4> <INPUT NAME=\"IID12*\"
SIZE=6> <INPUT NAME=\"Qty12*\" SIZE=1><BR>"
" <INPUT NAME=\"SP13*\" SIZE=4> <INPUT NAME=\"IID13*\"
SIZE=6> <INPUT NAME=\"Qty13*\" SIZE=1><BR>"
" <INPUT NAME=\"SP14*\" SIZE=4> <INPUT NAME=\"IID14*\"
SIZE=6> <INPUT NAME=\"Qty14*\" SIZE=1><BR>"
"Execution Status: Total:<BR>"
"</font></PRE><HR>"
"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Process\">"
"<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
"</FORM></HTML>"
);
}

```

```

else
{
    c += sprintf(szForm+c, "Warehouse: %6.6d District: %2.2d Date: ",
                pNewOrderData->w_id,
                pNewOrderData->d_id);

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

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

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

        for(i=0; i<pNewOrderData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c, "%6.6d %6.6d %-24s %2.2d %3.3d
%1.1s $%6.2f $%7.2f <BR>",
                        pNewOrderData->OL[i].ol_supply_w_id,
                        pNewOrderData->OL[i].ol_i_id,
                        pNewOrderData->OL[i].ol_i_name,
                        pNewOrderData->OL[i].ol_quantity,
                        pNewOrderData->OL[i].ol_stock,
                        pNewOrderData->OL[i].ol_brand_generic,
                        pNewOrderData->OL[i].ol_i_price,
                        pNewOrderData->OL[i].ol_amount );
        }
    }
}
else
{
    c += sprintf(szForm+c,
                "%Disc:<BR>"

```

```

                                "Order Number: %8.8d  Number of Lines:          W_tax:
D_tax:<BR> <BR>"
                                " Supp_W Item_Id Item Name          Qty Stock B/G Price
Amount<BR>"
                                , pNewOrderData->o_id);

                                i = 0;
                                }

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

                                if ( bValid )
                                    c += sprintf(szForm+c, "Execution Status: Transaction comitted.
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 += sprintf(szForm+c, "%2.2d-%2.2d-%4.4d %2.2d:%2.2d:%2.2d",
        pPaymentData->h_date.day,
        pPaymentData->h_date.month,
        pPaymentData->h_date.year,
        pPaymentData->h_date.hour,
        pPaymentData->h_date.minute,
        pPaymentData->h_date.second);
}

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

        , Term.pClientData[iTermId].w_id, pPaymentData->d_id
        , pPaymentData->w_street_1, pPaymentData->d_street_1

```



```

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

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

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

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

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

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

```

```

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

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

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

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

    if ( bInput )
    {
        strcpy(szForm+c,
            "District: <INPUT NAME=\"DID*\" SIZE=1><BR>"
            "Customer:  <INPUT  NAME=\"CID*\"  SIZE=4>                Name:"
            "<INPUT NAME=\"CLT*\" SIZE=23><BR>"
            "Cust-Balance:<BR> <BR>"
            "Order-Number:      Entry-Date:                Carrier-Number:<BR>"
            "Supply-W  Item-Id  Qty  Amount  Delivery-Date<BR> <BR> <BR>"
            "<BR> <BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
            "<BR></font></PRE>"
            "<HR><INPUT                TYPE=\"submit\"                NAME=\"CMD\""
            "VALUE=\"Process\"><INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"Menu\">"
            "</BODY></FORM></HTML>");
    }
    else
    {
        c += wsprintf(szForm+c,
            "District: %2.2d<BR>"
            "Customer: %4.4d  Name: %-16s %-2s %-16s<BR>",

```

```

        pOrderStatusData->d_id, pOrderStatusData->c_id,
        pOrderStatusData->c_first, pOrderStatusData->c_middle, pOrderStatusData-
>c_last);

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

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

        for(i=0; i< pOrderStatusData->o_ol_cnt; i++)
        {
            c += sprintf(szForm+c, " %6.6d %6.6d %2.2d $%8.2f %2.2d-
%2.2d-%4.4d<BR>",
            pOrderStatusData->OL[i].ol_supply_w_id,
            pOrderStatusData->OL[i].ol_i_id,
            pOrderStatusData->OL[i].ol_quantity,
            pOrderStatusData->OL[i].ol_amount,
            pOrderStatusData->OL[i].ol_delivery_d.day,
            pOrderStatusData->OL[i].ol_delivery_d.month,
            pOrderStatusData->OL[i].ol_delivery_d.year);
        }

        strncpy( szForm+c, szBR, (15-i)*5 );
        c += (15-i)*5;

        strcpy(szForm+c,
        "</font></PRE><HR><INPUT TYPE=\"submit\" NAME=\"CMD\"
VALUE=\"..NewOrder..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Order-
Status..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Stock-
Level..\">"
        "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
        "</BODY></FORM></HTML>" );
    }
}

/* FUNCTION: MakeDeliveryForm
*
* COMMENTS: The internal client buffer is created when the terminal id is assigned and

```

```

should not
*
*/
be freed except when the client terminal id is no longer needed.

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char
*szForm)
{
    int    c;

    c = wsprintf(szForm,
        "<HTML><HEAD><TITLE>TPC-C Delivery</TITLE></HEAD><BODY>"
        "<FORM ACTION=\"tpcc.dll\" METHOD=\"GET\">"
        "<INPUT TYPE=\"hidden\" NAME=\"STATUSID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"ERROR\" VALUE=\"0\">"
        "<INPUT TYPE=\"hidden\" NAME=\"FORMID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"TERMINID\" VALUE=\"%d\">"
        "<INPUT TYPE=\"hidden\" NAME=\"SYNCID\" VALUE=\"%d\">"
        "<PRE><font face=\"Courier\">                Delivery<BR>"
        "Warehouse: %6.6d<BR> <BR>",
        (!bInput    &&    (pDeliveryData->exec_status_code    !=    eOK))    ?
ERR_TYPE_DELIVERY_POST : 0,
        DELIVERY_FORM,    iTermId,    Term.pClientData[iTermId].iSyncId,
Term.pClientData[iTermId].w_id);

    if ( bInput )
    {
        strcpy( szForm+c,
            "Carrier Number: <INPUT NAME=\"OCD*\" SIZE=1><BR> <BR>"
            "Execution Status: <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR>"
            " <BR> <BR> <BR> <BR> <BR> <BR> <BR> <BR> <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> <BR> </font></PRE>"
            "<HR><INPUT                TYPE=\"submit\"                NAME=\"CMD\"
            VALUE=\"..NewOrder..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Payment..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Delivery..\">"
            "<INPUT    TYPE=\"submit\"    NAME=\"CMD\"    VALUE=\"..Order-
            Status..\">"
            "<INPUT    TYPE=\"submit\"    NAME=\"CMD\"    VALUE=\"..Stock-
            Level..\">"
            "<INPUT TYPE=\"submit\" NAME=\"CMD\" VALUE=\"..Exit..\">"
            "</BODY></FORM></HTML>"

```

```

        , pDeliveryData->o_carrier_id,
        (pDeliveryData->exec_status_code == eOK) ? "Delivery has been queued." :
"Delivery Post Failed "
    );
}
}

```

```

/* FUNCTION: ProcessNewOrderForm

```

```

*
* PURPOSE: This function gets and validates the input data from the new order form
*           filling in the required input variables. it then calls the SQLNewOrder
*           transaction, constructs the output form and writes it back to client
*           browser.
*/

```

```

void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)

```

```

{
    PNEW_ORDER_DATA pNewOrder;

    pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();

    ZeroMemory(pNewOrder, sizeof(NEW_ORDER_DATA));
    pNewOrder->w_id = Term.pClientData[iTermId].w_id;
    GetNewOrderData(pECB->lpszQueryString, pNewOrder);

    Term.pClientData[iTermId].pTxn->NewOrder();

    pNewOrder = Term.pClientData[iTermId].pTxn->BuffAddr_NewOrder();
    MakeNewOrderForm(iTermId, pNewOrder, OUTPUT_FORM, szBuffer);
}

```

```

/* FUNCTION: void ProcessPaymentForm

```

```

*
* PURPOSE: This function gets and validates the input data from the payment form
*           filling in the required input variables. It then calls the SQLPayment
*           transaction, constructs the output form and writes it back to client
*           browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK *pECB passed in structure
pointer from inetsrv.
*           int iTermId client
browser terminal id
*/

```

```

void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)

```

```

{
    PPAYMENT_DATA pPayment;

    pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
    ZeroMemory(pPayment, sizeof(PAYMENT_DATA));
    pPayment->w_id = Term.pClientData[iTermId].w_id;
}

```

```

    GetPaymentData(pECB->lpszQueryString, pPayment);

    Term.pClientData[iTermId].pTxn->Payment();

    pPayment = Term.pClientData[iTermId].pTxn->BuffAddr_Payment();
    MakePaymentForm(iTermId, pPayment, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessOrderStatusForm
 *
 * PURPOSE:   This function gets and validates the input data from the Order Status
 *            form filling in the required input variables. It then calls the
 *            SQLOrderStatus transaction, constructs the output form and writes it
 *            back to client browser.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK    *pECB passed in structure
 *            pointer from inetsrv.
 *            int                          iTermId client
 *            browser terminal id
 *
 */

void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)
{
    PORDER_STATUS_DATA    pOrderStatus;

    pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
    ZeroMemory(pOrderStatus, sizeof(ORDER_STATUS_DATA));
    pOrderStatus->w_id = Term.pClientData[iTermId].w_id;
    GetOrderStatusData(pECB->lpszQueryString, pOrderStatus);

    Term.pClientData[iTermId].pTxn->OrderStatus();

    pOrderStatus = Term.pClientData[iTermId].pTxn->BuffAddr_OrderStatus();
    MakeOrderStatusForm(iTermId, pOrderStatus, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessDeliveryForm
 *
 * PURPOSE:   This function gets and validates the input data from the delivery form
 *            filling in the required input variables. It then calls the PostDeliveryInfo
 *            Api, The client is then informed that the transaction has been posted.
 *
 * ARGUMENTS: EXTENSION_CONTROL_BLOCK    *pECB passed in structure
 *            pointer from inetsrv.
 *            int                          iTermId client
 *            browser terminal id
 *
 */

void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer)
{

```

```

char    *ptr = pECB->lpszQueryString;

PDELIVERY_DATA    pDelivery;

pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
ZeroMemory(pDelivery, sizeof(DELIVERY_DATA));
pDelivery->w_id = Term.pClientData[iTermId].w_id;

pDelivery->o_carrier_id =          GetIntKeyValue(&ptr,          "OCD*",
ERR_DELIVERY_MISSING_OCD_KEY, ERR_DELIVERY_CARRIER_INVALID);
if ( pDelivery->o_carrier_id > 10 || pDelivery->o_carrier_id < 1 )
    throw new CWEBCLNT_ERR( ERR_DELIVERY_CARRIER_ID_RANGE );

if (dwNumDeliveryThreads)
{
    //post delivery info
    if ( PostDeliveryInfo(pDelivery->w_id, pDelivery->o_carrier_id) )
        pDelivery->exec_status_code = eDeliveryFailed;
    else
        pDelivery->exec_status_code = eOK;
}
else // delivery is done synchronously if no delivery threads configured
    Term.pClientData[iTermId].pTxn->Delivery();

pDelivery = Term.pClientData[iTermId].pTxn->BuffAddr_Delivery();
MakeDeliveryForm(iTermId, pDelivery, OUTPUT_FORM, szBuffer);
}

/* FUNCTION: ProcessStockLevelForm
*
* PURPOSE:   This function gets and validates the input data from the Stock Level
*           form filling in the required input variables. It then calls the
*           SQLStockLevel transaction, constructs the output form and writes it
*           back to client browser.
*
* ARGUMENTS: EXTENSION_CONTROL_BLOCK    *pECB passed in structure
pointer from inetsrv.
*           int                          iTermId client
browser terminal id
*/

void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer)
{
    char    *ptr = pECB->lpszQueryString;

    PSTOCK_LEVEL_DATA    pStockLevel;

    pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
    ZeroMemory( pStockLevel, sizeof(STOCK_LEVEL_DATA) );

    pStockLevel->w_id = Term.pClientData[iTermId].w_id;

```

```

    pStockLevel->d_id = Term.pClientData[iTermId].d_id;

    pStockLevel->threshold = GetIntKeyValue(&ptr, "TT*",
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID);
    if ( pStockLevel->threshold >= 100 || pStockLevel->threshold < 0 )
        throw new CWEBCLNT_ERR( ERR_STOCKLEVEL_THRESHOLD_RANGE );

    Term.pClientData[iTermId].pTxn->StockLevel();

    pStockLevel = Term.pClientData[iTermId].pTxn->BuffAddr_StockLevel();
    MakeStockLevelForm(iTermId, pStockLevel, OUTPUT_FORM, szBuffer);
}

```

```

/* FUNCTION: GetNewOrderData

```

```

*
* PURPOSE: This function extracts and validates the new order form data from an http command
string.

```

```

* ARGUMENTS: LPSTR lpszQueryString client browser http
command string

```

```

* NEW_ORDER_DATA *pNewOrderData pointer
to new order data structure

```

```

*
*/

```

```

void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData)

```

```

{
    char szTmp[26];
    int i;
    short items;
    int ol_i_id, ol_quantity;
    char *ptr = lpszQueryString;

    static char szSP[MAX_OL_NEW_ORDER_ITEMS][6] =
        { "SP00*", "SP01*", "SP02*", "SP03*", "SP04*",
          "SP05*", "SP06*", "SP07*", "SP08*", "SP09*",
          "SP10*", "SP11*", "SP12*", "SP13*", "SP14*" };
    static char szIID[MAX_OL_NEW_ORDER_ITEMS][7] =
        { "IID00*", "IID01*", "IID02*", "IID03*", "IID04*",
          "IID05*", "IID06*", "IID07*", "IID08*", "IID09*",
          "IID10*", "IID11*", "IID12*", "IID13*", "IID14*" };
    static char szQty[MAX_OL_NEW_ORDER_ITEMS][7] =
        { "Qty00*", "Qty01*", "Qty02*", "Qty03*", "Qty04*",
          "Qty05*", "Qty06*", "Qty07*", "Qty08*", "Qty09*",
          "Qty10*", "Qty11*", "Qty12*", "Qty13*", "Qty14*" };

    pNewOrderData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_NEWORDER_FORM_MISSING_DID, ERR_NEWORDER_DISTRICT_INVALID);
    pNewOrderData->c_id = GetIntKeyValue(&ptr, "CID*",
ERR_NEWORDER_CUSTOMER_KEY, ERR_NEWORDER_CUSTOMER_INVALID);

    for(i=0, items=0; i<MAX_OL_NEW_ORDER_ITEMS; i++)

```



```

        {
            GetKeyValue(&ptr,          szSP[i],          szTmp,          sizeof(szTmp),
ERR_NEWORDER_MISSING_SUPPW_KEY);
            if ( szTmp[0] )
            {
                if ( !IsNumeric(szTmp) )
                    throw new
CWEBCLNT_ERR( ERR_NEWORDER_SUPPW_INVALID );
                pNewOrderData->OL[items].ol_supply_w_id = atoi(szTmp);

                ol_i_id = pNewOrderData->OL[items].ol_i_id =
                    GetIntKeyValue(&ptr,          szIID[i],
ERR_NEWORDER_MISSING_IID_KEY, ERR_NEWORDER_ITEMID_INVALID);
                if ( ol_i_id > 999999 || ol_i_id < 1 )
                    throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_RANGE );

                ol_quantity = pNewOrderData->OL[items].ol_quantity =
                    GetIntKeyValue(&ptr,          szQty[i],
ERR_NEWORDER_MISSING_QTY_KEY, ERR_NEWORDER_QTY_INVALID);
                if ( ol_quantity > 99 || ol_quantity < 1 )
                    throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_RANGE );

                items++;
            }
            else
            {
                // nothing entered for supply warehouse, so item id and qty must also be
blank
                GetKeyValue(&ptr,          szIID[i],          szTmp,          sizeof(szTmp),
ERR_NEWORDER_MISSING_IID_KEY);
                if ( szTmp[0] )
                    throw new
CWEBCLNT_ERR( ERR_NEWORDER_ITEMID_WITHOUT_SUPPW );

                GetKeyValue(&ptr,          szQty[i],          szTmp,          sizeof(szTmp),
ERR_NEWORDER_MISSING_QTY_KEY);
                if ( szTmp[0] )
                    throw new
CWEBCLNT_ERR( ERR_NEWORDER_QTY_WITHOUT_SUPPW );
            }
        }
        if ( items == 0 )
            throw new CWEBCLNT_ERR( ERR_NEWORDER_NOITEMS_ENTERED );

        pNewOrderData->o_ol_cnt = items;
    }

```

/* FUNCTION: GetPaymentData

*

* PURPOSE: This function extracts and validates the payment form data from an http command string.

*

```

* ARGUMENTS:      LPSTR          lpszQueryString      client  browser  http
command string
*                  PAYMENT_DATA    *pPaymentData    pointer  to
payment data structure
*/

```

```

void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData)
{
    char    szTmp[26];
    char    *ptr = lpszQueryString;
    BOOL    bCustIdBlank;
    int     iLen;

    pPaymentData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_PAYMENT_MISSING_DID_KEY, ERR_PAYMENT_DISTRICT_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        bCustIdBlank = TRUE;
        pPaymentData->c_id = 0;
    }
    else
    {
        // parse customer id and verify that last name was NOT entered
        bCustIdBlank = FALSE;
        if ( !IsNumeric(szTmp) )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_CUSTOMER_INVALID );
        pPaymentData->c_id = atoi(szTmp);
    }

    pPaymentData->c_w_id = GetIntKeyValue(&ptr, "CWI*",
ERR_PAYMENT_MISSING_CWI_KEY, ERR_PAYMENT_CWI_INVALID);
    pPaymentData->c_d_id = GetIntKeyValue(&ptr, "CDI*",
ERR_PAYMENT_MISSING_CDI_KEY, ERR_PAYMENT_CDI_INVALID);

    if ( bCustIdBlank )
    {
        // customer id is blank, so last name must be entered
        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new CWEBCLNT_ERR( ERR_PAYMENT_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(szTmp) > LAST_NAME_LEN )
            throw new
CWEBCLNT_ERR( ERR_PAYMENT_LAST_NAME_TO_LONG );

        strcpy(pPaymentData->c_last, szTmp);
        // pad with spaces so that the client layer doesn't have to do it
        // before passing parameters to stored procedure
        iLen = strlen(pPaymentData->c_last);

```

```

        memset(pPaymentData->c_last + iLen, '\0', LAST_NAME_LEN - iLen);
        pPaymentData->c_last[LAST_NAME_LEN] = 0;
    }
    else
    {
        // parse customer id and verify that last name was NOT entered
        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw new CWEBCLNT_ERR( ERR_PAYMENT_CID_AND_CLT );
    }

    GetKeyValue(&ptr, "HAM*", szTmp, sizeof(szTmp),
ERR_PAYMENT_MISSING_HAM_KEY);
    if (!IsDecimal(szTmp))
        throw new CWEBCLNT_ERR( ERR_PAYMENT_HAM_INVALID );
    pPaymentData->h_amount = atof(szTmp);
    if ( pPaymentData->h_amount >= 10000.00 || pPaymentData->h_amount < 0 )
        throw new CWEBCLNT_ERR( ERR_PAYMENT_HAM_RANGE );
}

```

/* FUNCTION: GetOrderStatusData

*
* PURPOSE: This function extracts and validates the payment form data from an http command string.
*
*/

```

void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData)
{
    char    szTmp[26];
    char    *ptr = lpszQueryString;
    int     iLen;

    pOrderStatusData->d_id = GetIntKeyValue(&ptr, "DID*",
ERR_ORDERSTATUS_MISSING_DID_KEY, ERR_ORDERSTATUS_DID_INVALID);

    GetKeyValue(&ptr, "CID*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CID_KEY);
    if ( szTmp[0] == 0 )
    {
        // customer id is blank, so last name must be entered
        pOrderStatusData->c_id = 0;
        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] == 0 )
            throw new
CWEBCLNT_ERR( ERR_ORDERSTATUS_MISSING_CID_CLT );

        _strupr( szTmp );
        if ( strlen(szTmp) > LAST_NAME_LEN )
            throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CLT_RANGE );

        strcpy(pOrderStatusData->c_last, szTmp);
        // pad with spaces so that the client layer doesn't have to do it
        // before passing parameters to stored procedure
    }
}

```

```

        iLen = strlen(pOrderStatusData->c_last);
        memset(pOrderStatusData->c_last + iLen, ' ', LAST_NAME_LEN - iLen);
        pOrderStatusData->c_last[LAST_NAME_LEN] = 0;
    }
    else
    {
        // parse customer id and verify that last name was NOT entered
        if ( !IsNumeric(szTmp) )
            throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_INVALID );
        pOrderStatusData->c_id = atoi(szTmp);
        GetKeyValue(&ptr, "CLT*", szTmp, sizeof(szTmp),
ERR_ORDERSTATUS_MISSING_CLT_KEY);
        if ( szTmp[0] != 0 )
            throw new CWEBCLNT_ERR( ERR_ORDERSTATUS_CID_AND_CLT );
    }
}
}

```

/* FUNCTION: BOOL IsNumeric(char *ptr)

*
* PURPOSE: This function determines if a string is numeric. It fails if any characters other
* than numeric and null terminator are present.
*
* ARGUMENTS: char *ptr pointer to string to check.
*
* RETURNS: BOOL FALSE if string is not all numeric
* TRUE if string contains only numeric characters
i.e. '0' - '9'
*/

BOOL IsNumeric(char *ptr)

```

{
    if ( *ptr == 0 )
        return FALSE;

    while( *ptr && isdigit(*ptr) )
        ptr++;
    return ( !*ptr );
}

```

/* FUNCTION: BOOL IsDecimal(char *ptr)

*
* PURPOSE: This function determines if a string is a non-negative decimal value.
* It fails if any characters other than a series of numbers followed by
* a decimal point, another series of numbers, and a null terminator are present.
*
* ARGUMENTS: char *ptr pointer to string to check.
*
* RETURNS: BOOL FALSE if string is not a valid non-negative decimal value
* TRUE if string is OK
*/

BOOL IsDecimal(char *ptr)

```

{

```

```

char *dotptr;
BOOL bValid;

if ( *ptr == 0 )
    return FALSE;

// find decimal point
dotptr = strchr( ptr, '.' );
if (dotptr == NULL)
    // no decimal point, so just check for numeric
    return IsNumeric(ptr);
*dotptr = 0; // temporarily replace decimal with a terminator

if ( *ptr != 0 )
    bValid = IsNumeric(ptr);
// string starts with decimal point
else if (*(dotptr+1) == 0)
    return FALSE; // nothing but a decimal point is bad
else
    bValid = TRUE;

if (*(dotptr+1) != 0)
    // check text after decimal point
    bValid &= IsNumeric(dotptr+1);

*dotptr = '.'; // replace decimal point
return bValid;
}

```

tpcc.def

LIBRARY TPCC.DLL

EXPORTS

```

GetExtensionVersion    @1
HttpExtensionProc      @2
TerminateExtension     @3

```

tpcc.h

```

/* FILE: TPCC.H
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 * Version 4.10.000 audited by Richard Gimarc, Performance Metrics,
3/17/99
 *

```

* PURPOSE: Header file for ISAPI TPCC.DLL, defines structures and functions used in the isapi tpcc.dll.

*
*/

//VERSION RESOURCE DEFINES

```
#define _APS_NEXT_RESOURCE_VALUE 101
#define _APS_NEXT_COMMAND_VALUE 40001
#define _APS_NEXT_CONTROL_VALUE 1000
#define _APS_NEXT_SYMED_VALUE 101
```

```
#define TP_MAX_RETRIES 50
```

//note that the welcome form must be processed first as terminal ids assigned here, once the //terminal id is assigned then the forms can be processed in any order.

```
#define WELCOME_FORM 1 //beginning form no term id assigned, form id
#define MAIN_MENU_FORM 2 //term id assigned main menu form id
#define NEW_ORDER_FORM 3 //new order form id
#define PAYMENT_FORM 4 //payment form id
#define DELIVERY_FORM 5 //delivery form id
#define ORDER_STATUS_FORM 6 //order status id
#define STOCK_LEVEL_FORM 7 //stock level form id
```

//This macro is used to prevent the compiler error unused formal parameter

```
#define UNUSEDPARAM(x) (x = x)
```

//This structure defines the data necessary to keep distinct for each terminal or client connection.

```
typedef struct _CLIENTDATA
{
    int iNextFree; //index of next free element or -1 if this entry in use.
    int w_id; //warehouse id assigned at welcome form
    int d_id; //district id assigned at welcome form

    int iSyncId; //synchronization id
    int iTickCount; //time of last access;

    CTPCC_BASE *pTxn;
} CLIENTDATA, *PCLIENTDATA;
```

```

//This structure is used to define the operational interface for terminal id support
typedef struct _TERM
{
    int                iNumEntries;                //total allocated
terminal array entries
    int                iFreeList;                //next
available terminal array element or -1 if none
    int                iMasterSyncId;            //synchronization
id
    CLIENTDATA *pClientData;                    //pointer to allocated
client data
} TERM;

typedef TERM *PTERM;                            //pointer to
terminal structure type

```

```

enum WEBERROR
{
    NO_ERR,
    ERR_COMMAND_UNDEFINED,
    ERR_D_ID_INVALID,
    ERR_DELIVERY_CARRIER_ID_RANGE,
    ERR_DELIVERY_CARRIER_INVALID,
    ERR_DELIVERY_MISSING_OCD_KEY,
    ERR_DELIVERY_THREAD_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_HTML_ILL_FORMED,
    ERR_INVALID_SYNC_CONNECTION,
    ERR_INVALID_TERMID,
    ERR_LOADDLL_FAILED,
    ERR_MAX_CONNECTIONS_EXCEEDED,
    ERR_MEM_ALLOC_FAILED,
    ERR_MISSING_REGISTRY_ENTRIES,
    ERR_NEWORDER_CUSTOMER_INVALID,
    ERR_NEWORDER_CUSTOMER_KEY,
    ERR_NEWORDER_DISTRICT_INVALID,
    ERR_NEWORDER_FORM_MISSING_DID,
    ERR_NEWORDER_ITEMID_INVALID,
    ERR_NEWORDER_ITEMID_RANGE,
    ERR_NEWORDER_ITEMID_WITHOUT_SUPPW,
    ERR_NEWORDER_MISSING_IID_KEY,
    ERR_NEWORDER_MISSING_QTY_KEY,
    ERR_NEWORDER_MISSING_SUPPW_KEY,
    ERR_NEWORDER_NOITEMS_ENTERED,
    ERR_NEWORDER_QTY_INVALID,
    ERR_NEWORDER_QTY_RANGE,
    ERR_NEWORDER_QTY_WITHOUT_SUPPW,
    ERR_NEWORDER_SUPPW_INVALID,
    ERR_NO_SERVER_SPECIFIED,
    ERR_ORDERSTATUS_CID_AND_CLT,
    ERR_ORDERSTATUS_CID_INVALID,
    ERR_ORDERSTATUS_CLT_RANGE,

```

```

ERR_ORDERSTATUS_DID_INVALID,
ERR_ORDERSTATUS_MISSING_CID_CLT,
ERR_ORDERSTATUS_MISSING_CID_KEY,
ERR_ORDERSTATUS_MISSING_CLT_KEY,
ERR_ORDERSTATUS_MISSING_DID_KEY,
ERR_PAYMENT_CDI_INVALID,
ERR_PAYMENT_CID_AND_CLT,
ERR_PAYMENT_CUSTOMER_INVALID,
ERR_PAYMENT_CWI_INVALID,
ERR_PAYMENT_DISTRICT_INVALID,
ERR_PAYMENT_HAM_INVALID,
ERR_PAYMENT_HAM_RANGE,
ERR_PAYMENT_LAST_NAME_TOO_LONG,
ERR_PAYMENT_MISSING_CDI_KEY,
ERR_PAYMENT_MISSING_CID_CLT,
ERR_PAYMENT_MISSING_CID_KEY,
ERR_PAYMENT_MISSING_CLT,
ERR_PAYMENT_MISSING_CLT_KEY,
ERR_PAYMENT_MISSING_CWI_KEY,
ERR_PAYMENT_MISSING_DID_KEY,
ERR_PAYMENT_MISSING_HAM_KEY,
ERR_STOCKLEVEL_MISSING_THRESHOLD_KEY,
ERR_STOCKLEVEL_THRESHOLD_INVALID,
ERR_STOCKLEVEL_THRESHOLD_RANGE,
ERR_VERSION_MISMATCH,
ERR_W_ID_INVALID

```

```
};
```

```
class CWEBCLNT_ERR : public CBaseErr
```

```
{
```

```
public:
```

```
CWEBCLNT_ERR(WEBERROR Err)
```

```
{
```

```

m_Error = Err;
m_szTextDetail = NULL;
m_SystemErr = 0;
m_szErrorText = NULL;

```

```
};
```

```
CWEBCLNT_ERR(WEBERROR Err, char *szTextDetail, DWORD dwSystemErr)
```

```
{
```

```

m_Error = Err;
m_szTextDetail = new char[strlen(szTextDetail)+1];
strcpy(m_szTextDetail, szTextDetail);
m_SystemErr = dwSystemErr;
m_szErrorText = NULL;

```

```
};
```

```
~CWEBCLNT_ERR()
```

```
{
```

```

if (m_szTextDetail != NULL)
    delete [] m_szTextDetail;

```



```

        if (m_szErrorText != NULL)
            delete [] m_szErrorText;
    };

    WEBERROR    m_Error;
    char        *m_szTextDetail;    //
    char        *m_szErrorText;
    DWORD       m_SystemErr;

    int ErrorType() {return ERR_TYPE_WEBDLL;};
    char *ErrorTypeStr() { return "WEBCLIENT"; }
    int ErrorNum() {return m_Error;};
    char *ErrorText();

};

//These constants have already been defined in engstut.h, but since we do
//not want to include it in the delisrv executable
#define TXN_EVENT_START        2
#define TXN_EVENT_STOP        4
#define TXN_EVENT_WARNING        6                //used to record a warning into the log

//function prototypes

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved);
void WriteMessageToEventLog(LPTSTR lpszMsg);
void ProcessQueryString(EXTENSION_CONTROL_BLOCK *pECB, int *pCmd, int *pFormId, int
*pTermId, int *pSyncId);
void WelcomeForm(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void SubmitCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void BeginCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void ProcessCmd(EXTENSION_CONTROL_BLOCK *pECB, int iFormId, int iTermId);
void StatsCmd(EXTENSION_CONTROL_BLOCK *pECB, char *szBuffer);
void ErrorMessage(EXTENSION_CONTROL_BLOCK *pECB, int iError, int iErrorType, char
*szMsg, int iTermId);
void GetKeyValue(char **pQueryString, char *pKey, char *pValue, int iMax, WEBERROR err);
int  GetIntKeyValue(char **pQueryString, char *pKey, WEBERROR NoKeyErr, WEBERROR
NotIntErr);
void TermInit(void);
void TermDeleteAll(void);
int  TermAdd(void);
void TermDelete(int id);
void ErrorForm(EXTENSION_CONTROL_BLOCK *pECB, int iType, int iErrorNum, int iTermId,
int iSyncId, char *szErrorText, char *szBuffer );
void MakeMainMenuForm(int iTermId, int iSyncId, char *szForm);
void MakeStockLevelForm(int iTermId, STOCK_LEVEL_DATA *pStockLevelData, BOOL bInput,
char *szForm);
void MakeNewOrderForm(int iTermId, NEW_ORDER_DATA *pNewOrderData, BOOL bInput, char
*szForm);
void MakePaymentForm(int iTermId, PAYMENT_DATA *pPaymentData, BOOL bInput, char
*szForm);
void MakeOrderStatusForm(int iTermId, ORDER_STATUS_DATA *pOrderStatusData, BOOL bInput,
char *szForm);

```

```

void MakeDeliveryForm(int iTermId, DELIVERY_DATA *pDeliveryData, BOOL bInput, char
*szForm);
void ProcessNewOrderForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void ProcessPaymentForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessOrderStatusForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void ProcessDeliveryForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char *szBuffer);
void ProcessStockLevelForm(EXTENSION_CONTROL_BLOCK *pECB, int iTermId, char
*szBuffer);
void GetNewOrderData(LPSTR lpszQueryString, NEW_ORDER_DATA *pNewOrderData);
void GetPaymentData(LPSTR lpszQueryString, PAYMENT_DATA *pPaymentData);
void GetOrderStatusData(LPSTR lpszQueryString, ORDER_STATUS_DATA *pOrderStatusData);
BOOL PostDeliveryInfo(long w_id, short o_carrier_id);
BOOL IsNumeric(char *ptr);
BOOL IsDecimal(char *ptr);
void DeliveryWorkerThread(void *ptr);

```

tpcc.rc

```

//Microsoft Developer Studio generated resource script.
//
#include "resource.h"

#define APSTUDIO_READONLY_SYMBOLS
//
// Generated from the TEXTINCLUDE 2 resource.
//
#include "afxres.h"

//
// English (U.S.) resources

#if !defined(AFX_RESOURCE_DLL) || defined(AFX_TARG_ENU)
#ifdef _WIN32
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
#pragma code_page(1252)
#endif

#ifdef _MAC
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 0,4,0,0
PRODUCTVERSION 0,4,0,0

```

```

FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
FILEFLAGS 0x1L
#else
FILEFLAGS 0x0L
#endif
FILEOS 0x40004L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
  BLOCK "StringFileInfo"
  BEGIN
    BLOCK "040904b0"
    BEGIN
      VALUE "Comments", "TPC-C HTML DLL Server (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);

    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 );
    if (FAILED(hr))
        throw new CCOMERR( hr );
}

void CTPCC_COM::Payment()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pPayment->Payment(m_vTxn, &vTxn_out);

    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 );
    if (FAILED(hr))
        throw new CCOMERR( hr );
}

void CTPCC_COM::StockLevel()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pStockLevel->StockLevel(m_vTxn, &vTxn_out);

    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 );
    if (FAILED(hr))
        throw new CCOMERR( hr );
}

void CTPCC_COM::OrderStatus()
{
    VARIANT vTxn_out;

    HRESULT hr = m_pOrderStatus->OrderStatus(m_vTxn, &vTxn_out);

    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 );
    if (FAILED(hr))
        throw new CCOMERR( hr );
}

```


tpcc_com.h

```
/* FILE: TPCC_COM.H
 * Microsoft TPC-C Kit Ver. 4.20.000
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *
 * not yet audited
 *
 * PURPOSE: Header file for TPC-C COM+ class implementation.
 *
 * Change history:
 * 4.20.000 - first version
 */
```

```
#pragma once
```

```
#include <stdio.h>
```

```
#include "..\..\tpcc_com_ps\src\tpcc_com_ps.h"
```

```
// need to declare functions for import, unless define has already been created
```

```
// by the DLL's .cpp module for export.
```

```
#ifndef DllDecl
```

```
#define DllDecl __declspec( dllimport )
```

```
#endif
```

```
class CCOMERR : public CBaseErr
```

```
{
```

```
private:
```

```
char m_szErrorText[64];
```

```
public:
```

```
// use this interface for genuine COM errors
```

```
CCOMERR( HRESULT hr )
```

```
{
```

```
    m_hr = hr;
```

```
    m_iErrorType = 0;
```

```
    m_iError = 0;
```

```
}
```

```
// use this interface to impersonate a non-COM error type
```

```
CCOMERR( int iErrorType, int iError )
```

```
{
```

```
    m_iErrorType = iErrorType;
```

```
    m_iError = iError;
```

```
    m_hr = S_OK;
```

```
}
```

```
int m_hr;
```

```
int m_iErrorType;
```

```
int m_iError;
```

```

// A CCOMERR class can impersonate another class, which happens if the error
// was not actually a COM Services error, but was simply transmitted back via COM.
int ErrorType()
{
    if (m_iErrorType == 0)
        return ERR_TYPE_COM;
    else
        return m_iErrorType;
}

char *ErrorTypeStr() { return "COM"; }

int ErrorNum()
{
    if (m_iErrorType == 0)
        return m_hr;           // return COM error
    else
        return m_iError; // return impersonated error
}

char *ErrorText()
{
    if (m_hr == S_OK)
        sprintf( m_szErrorText, "Error: Class %d, error # %d",
m_iErrorType, m_iError );
    else
        sprintf( m_szErrorText, "Error: COM HRESULT %x", m_hr );
    return m_szErrorText;
}
};

class DllDecl CTPCC_COM : public CTPCC_BASE
{
private:
    BOOL m_bSinglePool;

    // COM Interface pointers
    ITPCC*      m_pNewOrder;
    ITPCC*      m_pPayment;
    ITPCC*      m_pStockLevel;
    ITPCC*      m_pOrderStatus;

    struct COM_DATA
    {
        int ErrorType;
        int error;
        union
        {
            NEW_ORDER_DATA      NewOrder;
            PAYMENT_DATA         Payment;
            DELIVERY_DATA        Delivery;
            STOCK_LEVEL_DATA     StockLevel;
            ORDER_STATUS_DATA    OrderStatus;
        }
    };
};

```

```

        } u;
    } *m_pTxn;

    VARIANT m_vTxn;
public:
    CTPCC_COM(BOOL bSinglePool);
    ~CTPCC_COM(void);

    inline PNEW_ORDER_DATA      BuffAddr_NewOrder()      { return
&m_pTxn->u.NewOrder; };
    inline PPAYMENT_DATA        BuffAddr_Payment()        { return
&m_pTxn->u.Payment; };
    inline PDELIVERY_DATA       BuffAddr_Delivery()       { return
&m_pTxn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA     BuffAddr_StockLevel()     { return  &m_pTxn-
>u.StockLevel; };
    inline PORDER_STATUS_DATA   BuffAddr_OrderStatus()   { return  &m_pTxn-
>u.OrderStatus; };

    void NewOrder      ();
    void Payment       ();
    void StockLevel    ();
    void OrderStatus();
    void Delivery      () { throw new CCOMERR(E_NOTIMPL); } // not
supported
};

inline void ReleaseInterface(IUnknown *pUnk)
{
    if (pUnk)
    {
        pUnk->Release();
        pUnk = NULL;
    }
}

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_COM* CTPCC_COM_new(BOOL);

typedef CTPCC_COM* (TYPE_CTPCC_COM)(BOOL);

```

client_utils.c

```

/* client_utils.c
*/

#include <stdio.h>
#include <time.h>
#include <windows.h>
#include <winperf.h>

```

```

#include <winsock.h>
#include "client_utils.h"

#define Li2Double(x) ((double)((x).HighPart) * 4.294967296E9 + (double)((x).LowPart))

static LARGE_INTEGER pFreq;
static double sFreq;
static int print_thread_id = 1;
static int user_id = 0;
static char *user_code = "C";

/*
 * get_thread_id
 *   A function that returns the thread ID of the current thread
 */
static int get_thread_id()
{
    return(GetCurrentThreadId());
}

/*
 * get_prefix
 *   Format the output prefix for printing:
 *   It contains the user_id, 'C' or 'T' depending on whether it
 *   is a terminal or a client and optional a thread identifier
 *   The prefix is written in the buffer passed in by the caller.
 */
static void get_prefix(char *buffer)
{
    if (print_thread_id) {
        int thread_id = get_thread_id();
        sprintf(buffer, "%s(%d-%s-%d)%s",
            user_id < 10 ? " " : user_id < 100 ? " " : "",
            user_id,
            user_code,
            thread_id,
            thread_id < 10 ? " " : "");
    } else {
        sprintf(buffer, "%s(%2d-%s)",
            user_id < 10 ? " " : "", user_id, user_code);
    }
}

/*
 * err_printf
 *   A var-arg function that appends the current time and
 *   other data to the print request and sends it to stderr
 *   if it is not a web client, to a file if it is
 */
void err_printf(char *format, ...)
{
    time_t cur_timet;

```

```

char time_str[30];
char line_prefix[50];
va_list ap;

va_start(ap, format);

cur_timet = time(&cur_timet);
strftime(time_str, 29, "%X", localtime(&cur_timet));

get_prefix(line_prefix);

fprintf(ERROUT, "%s %s - ", line_prefix, time_str);
vfprintf(ERROUT, format, ap);
fflush(ERROUT);

va_end(ap);
}

/*
 * encina_error_message
 *
 * Report an encina error message by interpreting it and writing
 * it to both the logfile (if any) and to standard error
 */
void encina_error_message(char *msg, unsigned long n)
{
    char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
    encina_StatusToString(n, ENCINA_MAX_STATUS_STRING_SIZE, errorMsg);
    err_printf("ERROR: %s. Error code = %s (%d 0x%x) \n", msg, errorMsg, n, n);
}

int get_time_init()
{
    QueryPerformanceFrequency(&pFreq);
    sFreq=Li2Double(pFreq);
    return 0;
}

int get_local_time(time_type *timeP)
{
    double cur_t;
    LARGE_INTEGER counter;

    QueryPerformanceCounter(&counter);
    cur_t = Li2Double(counter) / sFreq;
    timeP->sec = (long)cur_t;
    /* timeP->usec = ((long)cur_t - timeP->sec) * 1000000;*/
    timeP->usec = (long)((cur_t - timeP->sec) * 1000000);
    return 0;
}

/*

```

```

* time_diff_ms
*   Return the difference in milliseconds between two times
*/
int time_diff_ms(struct timeval *t2, struct timeval *t1)
{
    int t_diff;

    t_diff = (t2->tv_usec + 1000000 - t1->tv_usec + 500) / 1000 +
        (t2->tv_sec - t1->tv_sec - 1) * 1000;

    return(t_diff);
}

/*
* perfClntDataInit:
*   Initialization for the shared file mapping.
*
* return: pointer to the shared memory space
*
* This routine creates a named mapped memory section that is used
* to communicate the TPCC performance data to the extensible
* counter DLL for NT perfmon.
*/
total_tran_count_t *perfClntDataInit()
{
    HANDLE hMappedObject;
    total_tran_count_t *pClntInfo = NULL;
    TCHAR szMappedObjectName[] = TEXT("TPCC_CLNT_COUNTER_BLOCK");

    /* create named section for the performance data */
    hMappedObject = CreateFileMapping((HANDLE)0xFFFFFFFF,
        NULL,
        PAGE_READWRITE,
        0,
        sizeof(total_tran_count_t),
        szMappedObjectName);
    if (hMappedObject == NULL) {
        err_printf("perfClntDataInit: CreateFileMapping failed %x\n",
            GetLastError());
        pClntInfo = NULL;
    } else {
        /* map the section and assign the counter block pointer
        * to this section of memory
        */
        pClntInfo = (total_tran_count_t *) MapViewOfFile(hMappedObject,
            FILE_MAP_ALL_ACCESS,
            0,
            0,
            0);
        if (pClntInfo == NULL) {
            err_printf("perfClntDataInit: MapViewOfFile failed %x\n",
                GetLastError());
        }
    }
}

```

```

        else {
            err_printf("perfClntDataInit: MapViewOfFile success \n");
        }
    }

    return(pClntInfo);
}

```

client_utils.h

```

#ifndef TPCC_CLIENT_UTILS_H
#define TPCC_CLIENT_UTILS_H

#include <stdio.h>
#include <time.h>
#include <dce/rpc.h>
#include <dce/dce_error.h>
#include <encina/encina.h>
#include <stdlib.h>
#include <utils/trace.h>
#include <winsock.h>
#include "mon_client.h"
#include "../include/tpcc_type.h"

extern FILE * errtpcc;
extern FILE * logtpcc;
extern int debug;
extern char log_file_name[];
extern void logprintf( char *format, ...);
extern void err_printf( char *format, ...);
extern void encina_error_message(char *msg, unsigned long n);
extern int time_diff_ms(struct timeval *t2, struct timeval *t1);

typedef struct {
    int num;
    int errs;
    double RTtotal[2]; // 1 for server RT and 0 for client RT
    int RTcount;
} tran_info_t;

/*
 * total_tran_count_t
 *
 * structure that holds the total count of transaction of each type
 * as well as the reposne times.
 *
 */
typedef struct {
    tran_info_t tran[MAX_TRAN_TYPE + 1];
    int errors;
    double time;
} total_tran_count_t;

```

```

/* enc_status_t
 * structure that holds error information
 */
typedef struct {
    int status;
    int line;
    char file[268];
    unsigned long encinaError;
    char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
} enc_status_t;

#define FALSE 0
#define TRUE 1

#define DPRINT(args) if (0) err_printf args

#define CHECK_ENVIRON(str,var) if (str == NULL) { fprintf(ERROROUT, \
    "%s environment variable is not defined.\n",var); }

#define CHK_STATUS(st, val, _errMsg) \
    if(st) { \
        enc_status.status=val; \
        strcpy(enc_status.file, __FILE__); \
        enc_status.line= __LINE__; \
        enc_status.encinaError = st; \
        if(_errMsg)strcpy(enc_status.errorMsg, _errMsg); \
        if(st!=1) return; \
    }

#define UTIL_IDENT(a) a

#if ENCINA_C_ANSI_STRING_TOKEN_SUPPORT
#define UTIL_STRING(a) # a
#define UTIL_CONCAT(a, b) a ## b
#else /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */
#define UTIL_STRING(a) "a"
#define UTIL_CONCAT(a, b) UTIL_IDENT(a)b
#endif /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */

/* ENCINA_CALL: Make fail-fast calls on the various services. */
#define ENCINA_CALL(proc_name,call) \
{ \
    unsigned long _status; \
    ENCINA_CALL_RC(proc_name,call,_status); \
    if (_status) exit_program(_status); \
}

#define ENCINA_CALL_RC(proc_name,call,rc) \
{ \
    char _errorMsg[ENCINA_MAX_STATUS_STRING_SIZE]; \

```



```

DPRINT(("ENCINA_CALL_RC: before call %s\n", proc_name));          \
rc = (call);                                                       \
DPRINT(("ENCINA_CALL_RC: after call %s\n", proc_name));          \
if (rc) {                                                           \
    encina_StatusToString(rc, ENCINA_MAX_STATUS_STRING_SIZE,     \
        _errorMsg);                                               \
    err_printf( "%x\n", rc);                                       \
    err_printf( "%s\n", _errorMsg);                                 \
    err_printf( "%s\n", proc_name);                                 \
}                                                                     \
}

void err_printf(char *format, ...);
void encina_error_message(char *msg, unsigned long n);
int get_time_init();
int get_local_time(time_type *timeP);
int time_diff_ms(struct timeval *t2, struct timeval *t1);

#endif /* TPCC_CLIENT_UTILS_H */

```

mon_client.c

```

/*
 *      mon_client.c
 *
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <time.h>
#include <pthread.h>
#include <tpm/mon/mon.h>
#include <utils/trace.h>
#include "../include/delivery.h"
#include "../include/neworder.h"
#include "../include/payment.h"
#include "../include/stocklevel.h"
#include "../include/orderstatus.h"
#include "../include/tpcc_type.h"
#include "mon_client.h"
#include "client_utils.h"

extern total_tran_count_t *perfCntDataInit();
static void read_mon_environment(void);

static char *cellName;
static int envRetrieval = 0;
static int useSecurity = FALSE;
static CRITICAL_SECTION  init_lock;
static total_tran_count_t *pClientInfo=NULL; /* keep stats for the client process */

```

```

static num_active_threads = 0;
static int iStatsFrequency = 1;
FILE *errtpcc;
char *errFile = "C:/temp/tpcc_encina.out";
enc_status_t enc_status;

#define NewOrder_code  NEWO_TRANS
#define Payment_code   PAYMENT_TRANS
#define OrderStatus_code ORDER_STAT_TRANS
#define Delivery_code  DELIVERY_TRANS
#define StockLevel_code STOCK_TRANS

#define INT_ENV_VALUE(var, default) \
    (var = getenv(#var) ? atoi(getenv(#var)) : default)

#define PRE_RPC_WORK(headerP, tran, sub_tran) \
    if (iStatsFrequency > 0) \
        pre_rpc(headerP, tran, sub_tran); \
    else \
        (headerP)->stats = 0;
#define POST_RPC_WORK(headerP, tran) \
    if (iStatsFrequency > 0) \
        post_rpc(headerP, tran)

/* CALTPCC
 * Macro to sends 1 RPC and then handles any errors.
 *
 * The macro takes the name of the RPC (e.g., NewOrder)
 * and makes the RPC by calling the appropriate function
 * (e.g., impTPCCNewOrder).
 */
#define CALLTPCC(name,length,dataP,header,trpcStatusP) \
{ \
    UTIL_CONCAT(impTPCC,name)(length,dataP,&header,trpcStatusP); \
    if (*(trpcStatusP)) { \
        char msg[100]; \
        sprintf(msg, "TRPC error during impTPCC%s", UTIL_STRING(name)); \
        header.returncode = TRPC_ERROR; \
        encina_error_message(msg, *(trpcStatusP)); \
    } else if ((header.returncode != TPCC_SUCCESS) && \
        (header.returncode != INVALID_NEWO)) { \
        char msg[100]; \
        sprintf(msg, "App error during impTPCC%s: ", UTIL_STRING(name)); \
        encina_error_message(msg, header.returncode); \
    } \
}

/*
 * pre_rpc -- For debug purposes
 *
 * Called before an RPC is made.
 * Set the state of the thread and keep track of the time the RPC is sent.

```

```

* This is used by the Background thread to report the state of the client.
*/
static void pre_rpc(data_header *headerP,
                  int tran_type,
                  int sub_tran_type)
{
    if (iStatsFrequency < 1) {
        headerP->stats = 0;
    } else {
        int num;
        num = ++ (pClientInfo->tran[tran_type].num);
        headerP->stats = (num % iStatsFrequency==0) ? 1 : 0;
        if (headerP->stats)
            { /* measure the time for RT */
                get_local_time(&headerP->clnt_start);
                headerP->srv_start.sec = 0; /* initialize the server time */
                headerP->srv_start.usec = 0;
                headerP->srv_end.sec = 0;
                headerP->srv_end.usec = 0;
            }
    }
}

/*
* post_rpc
*
* Called when the RPC returns from the server
*
* Keeps track of the client response time and the server response time
* as well as the state of the thread. This is used by the background
* debug thread to report the state of the client
*/
static void post_rpc(data_header *headerP,
                   int tran_type)
{
    double time_diff;
    int tran_failed;
    struct timeval start_time, end_time;

    if (headerP->stats)
        get_local_time(&headerP->clnt_end);
    else
        return;

    /* Store the info for each client.
    * Note: Since we don't use mutex for performance reason, pClientInfo
    * may not be accurate if more than one thread work on the same
    * data at a same time. But this should give us reasonable info.
    */
    if ((headerP->returncode == TPCC_SUCCESS) ||
        (headerP->returncode == INVALID_NEWO)) {
        tran_failed = 0;
    } else {

```

```

pClientInfo->tran[tran_type].errs ++;
pClientInfo->errors ++;
tran_failed = 1;
}
if (headerP->stats && tran_type <= MAX_TRAN_TYPE && tran_type > 0
    && !tran_failed) {
    /* update total server round trip response time */
    start_time.tv_sec = headerP->srv_start.sec;
    start_time.tv_usec = headerP->srv_start.usec;
    end_time.tv_sec = headerP->srv_end.sec;
    end_time.tv_usec = headerP->srv_end.usec;
    time_diff = time_diff_ms(&end_time, &start_time);
    pClientInfo->tran[tran_type].RTtotal[1] += time_diff;
    DPRINTF(("srv start_time %d.%d, end_time %d.%d, time_diff %f\n",
            start_time.tv_sec, start_time.tv_usec,
            end_time.tv_sec, end_time.tv_usec,
            time_diff));

    /* update total client round trip response time */
    start_time.tv_sec = headerP->clnt_start.sec;
    start_time.tv_usec = headerP->clnt_start.usec;
    end_time.tv_sec = headerP->clnt_end.sec;
    end_time.tv_usec = headerP->clnt_end.usec;
    time_diff = time_diff_ms(&end_time, &start_time);
    pClientInfo->tran[tran_type].RTtotal[0] += time_diff;
    DPRINTF(("clnt start_time %d.%d, end_time %d.%d, time_diff %f\n",
            start_time.tv_sec, start_time.tv_usec,
            end_time.tv_sec, end_time.tv_usec,
            time_diff));

    /* update num for the number of trans which have RT measured */
    pClientInfo->tran[tran_type].RTcount ++;
}
}

/*
 * The following send_*** functions are called from CTPCC_ENCINA class.
 */

/*
 * send_new_order
 *     Send a new order request to the server
 */
int send_new_order(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, NEWO_TRANS, 0);
    CALLTPCC(NewOrder,length,dataP,header,&trpcStatus)
    POST_RPC_WORK(&header, NEWO_TRANS);
}

```

```

    if (header.returncode == INVALID_NEWO)
        return TPCC_SUCCESS;
    else
        return header.returncode;
}

/*
 * send_payment
 *     Send a payment request to the server
 */
int send_payment(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, PAYMENT_TRANS, 0);
    CALLTPCC(Payment,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, PAYMENT_TRANS);
    return header.returncode;
}

/*
 * send_order_status
 *     Send a order status request to the server
 */
int send_order_status(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, ORDER_STAT_TRANS, 0);
    CALLTPCC(OrderStatus,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, ORDER_STAT_TRANS);
    return header.returncode;
}

/*
 * send_delivery
 *     Send a delivery request to the server
 */
int send_delivery(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, DELIVERY_TRANS, 0);
    CALLTPCC(Delivery,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, DELIVERY_TRANS);
    return header.returncode;
}

/*
 * send_stock_level

```

```

*      Send a stock level request to the server
*/
int send_stock_level(long length, unsigned char *dataP)
{
    trpc_status_t trpcStatus;
    data_header header;

    PRE_RPC_WORK(&header, STOCK_TRANS, 0);
    CALLTPCC(StockLevel,length,dataP,header,&trpcStatus);
    POST_RPC_WORK(&header, STOCK_TRANS);
    return header.returncode;
}

/*
* Enroll the client:
*     get the necessary handles.
* This function should be called only once. Use static var client_enrolled to control it.
*/
void enroll_client()
{
    static char *clientName="tpcc_client";
    unsigned long status ;
    static int  client_enrolled = 0;
    unsigned32  client_authnLevel;
    unsigned32  client_authzSvc;
    time_type a_time;
    char err_msg[100];

    MUTEX_INIT(&init_lock);
    get_local_time(&a_time);
    srand(a_time.sec ^ a_time.usec);

    MUTEX_LOCK(&init_lock);
    if (client_enrolled) {
        MUTEX_UNLOCK(&init_lock);
        return;
    }

    /* open output file for tracing */
    errtpcc = fopen(errFile, "w");
    if(!errtpcc)
    {
        sprintf(err_msg, "Cannot open file %s", errFile);
        CHK_STATUS(1, ERROUT_FILE_NOT_FOUND,err_msg);
    }

    get_time_init();
    // initialize the space for perfmon
    pClientInfo = perfCIntDataInit();
    if (pClientInfo == NULL) // in case something wrong
        pClientInfo = malloc(sizeof(total_tran_count_t));
}

```

```

memset(pClientInfo, 0, sizeof(total_tran_count_t));

read_mon_environment();

if(!cellName)
    CHK_STATUS(30, CELL_NAME_UNAVAILABLE,
        "ENCINA_TPM_CELL is not set!");

if (useSecurity) {
    client_authnLevel = rpc_c_protect_level_connect;
    client_authzSvc = rpc_c_authz_dce;
} else {
    client_authnLevel = rpc_c_protect_level_none;
    client_authzSvc = rpc_c_authz_none;
}

if (envRetrieval == 0) {
    ENCINA_CALL_RC("mon_RetrieveEnable", mon_RetrieveEnable(FALSE), status);
    CHK_STATUS(status, MON_RETRIEVEENABLE_FAILED,
        "mon_RetrieveEnable failed");
}

err_printf("enroll_client: calling mon_InitClient \n");
ENCINA_CALL_RC("mon_InitClient", mon_InitClient(clientName, cellName), status);
CHK_STATUS(status, MON_INITCLIENT_FAILED,
    "mon_InitClient failed");

DPRINT(("mon_SecuritySetDefaults-> authn %d, authz %d\n",
    client_authnLevel, client_authzSvc));
ENCINA_CALL_RC("mon_SecuritySetDefaults",
    mon_SecuritySetDefaults(client_authnLevel, client_authzSvc),
    status);
CHK_STATUS(status, MON_SECURITYSET_FAILED,
    "mon_SecuritySetDefaults failed");

ENCINA_CALL_RC("mon_SetHandleCacheRefreshInterval",
    mon_SetHandleCacheRefreshInterval(300), status);
CHK_STATUS(status, MON_SETREFRESHINTERVAL_FAILED,
    "mon_SetHandleCacheRefreshInterval failed");

{
    dbInfo_data_t data;
    trpc_status_t trpcStatus;
    /* Get DB Info -- currently id does not do anything
    but it will tell us if there is a server out there.
    Better to know instead of when all the terminals
    are up and ready
    */
    impTPCCNOInfo(&data, &trpcStatus);
    if (trpcStatus) {
        char msg[100];
        sprintf(msg, "TRPC error during db info at init.");
        encina_error_message(msg, trpcStatus);
    }
}

```

```

        CHK_STATUS(33,NOINFO_TRPC_ERROR,
            "TRPC error during db info at init");
    }
}

client_enrolled = 1;
MUTEX_UNLOCK(&init_lock);
err_printf("end of enroll_client\n");
}

/*-----*/
/*   Read environment paramaters and registry entries   */
/*-----*/
static void read_mon_environment()
{
    char *env_str;
    char *registryKey = "SOFTWARE\\TransarcCorporation\\TxTpcc";
    HKEY hKey;
    DWORD size;
    DWORD type;
    char szTmp[256];

    cellName = getenv("ENCINA_TPM_CELL");
    CHECK_ENVIRON(cellName, "ENCINA_TPM_CELL");

    if (env_str = getenv("TPCC_ENV_RETRIEVE")) {
        envRetrieval = atoi(env_str);
    }

    if ( RegOpenKeyEx(HKEY_LOCAL_MACHINE, registryKey, 0, KEY_READ, &hKey) !=
        ERROR_SUCCESS )
        return;

    size = sizeof(szTmp);
    if ( RegQueryValueEx(hKey, "StatsFrequency", 0, &type, szTmp, &size)==ERROR_SUCCESS)
        iStatsFrequency = atoi(szTmp);

    RegCloseKey(hKey);
}

```

mon_client.h

```

/*
 *   mon_client.h
 *
 */

#ifndef MON_CLIENT_H
#define MON_CLIENT_H

#define MUTEX_T CRITICAL_SECTION

```



```

#define MUTEX_LOCK(a) EnterCriticalSection(a)
#define MUTEX_UNLOCK(a) LeaveCriticalSection(a)
#define MUTEX_INIT(mut) InitializeCriticalSection(mut)
#define MUTEX_DESTROY(mut) DeleteCriticalSection(mut)
#define ERROUT errtpcc

/*initialization status */
#define INIT_SUCCESS 0
#define INIT_FAILED 1
#define CELL_NAME_UNAVAILABLE 2
#define MON_RETRIEVEENABLE_FAILED 3
#define MON_INITCLIENT_FAILED 4
#define MON_SECURITYSET_FAILED 5
#define MON_SETREFRESHINTERVAL_FAILED 6
#define NOINFO_TRPC_ERROR 7
#define ENROLL_CLIENT_EXCEPTION 8
#define ERROUT_FILE_NOT_FOUND 9
#define LOG_FILE_NOT_FOUND 10
#define TPCC_KEY_NOT_FOUND 11
#define TERM_ALLOC_FAILED 12

/*
 * Routines and declarations that are common to all clients
 */
#if defined(__cplusplus)
extern "C" {
#endif
int send_new_order(long, unsigned char *);
int send_payment(long, unsigned char *);
int send_order_status(long, unsigned char *);
int send_delivery(long, unsigned char *);
int send_stock_level(long, unsigned char *);
void enroll_client();
#if defined(__cplusplus)
}
#endif

#endif /* MON_CLIENT_H */

```

tpcc_enc.cpp

```

// ctppc_enc.cpp: implementation of the CTPCC_ENCINA class.
//
////////////////////////////////////

#include <windows.h>
#include <process.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>

#ifdef ICECAP
#include <icapexp.h>
#endif

// need to declare functions for export
#define DllDecl __declspec( dllexport )

#include "..\..\common\src\trans.h" //tpckit transaction header contains definitions of structures
specific to TPC-C
#include "..\..\common\src\error.h"
#include "..\..\common\src\txn_base.h"
#include "tpcc_enc.h"
#include "..\include\tpcc_type.h"
#include "mon_client.h"
#include "client_utils.h"

static CRITICAL_SECTION TpCriticalSection;
extern "C" char *errFile;

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

            DisableThreadLibraryCalls(hModule);
            InitializeCriticalSection(&TpCriticalSection);
            break;

        case DLL_PROCESS_DETACH:
            DeleteCriticalSection(&TpCriticalSection);
            break;

        default:
            /* nothing */;
    }
    return TRUE;
}

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_ENCINA* CTPCC_ENCINA_new()
{
    return new CTPCC_ENCINA();
}

// wrapper routine for enroll_client
__declspec(dllexport) CTPCC_ENCINA* CTPCC_ENCINA_post_init()

```

```

{
    enroll_client();
    return NULL;
}

// constructor and destructor
CTPCC_ENCINA::CTPCC_ENCINA()
{
    // Add initialization of ENCINA Structures if any
    m_txn = (ENC_DATA *)malloc(sizeof(ENC_DATA));
    if (m_txn == NULL)
        throw new CENCERR(ERR_TYPE_MEMORY, ERR_FATAL_LEVEL);
}

CTPCC_ENCINA::~CTPCC_ENCINA()
{
    // free the data structure allocated with tmalloc
    free((char *)m_txn);
}

void CTPCC_ENCINA::NewOrder()
{
    // question: if we need to prepare the data?
    if (send_new_order(sizeof(ENC_DATA), (unsigned char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn->ErrorType, m_txn->error );
}

void CTPCC_ENCINA::Payment()
{
    if (send_payment(sizeof(ENC_DATA), (unsigned char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn->ErrorType, m_txn->error );
}

void CTPCC_ENCINA::Delivery()
{
    // Note: Delivery txn code in the tuxedo server does not implement logging of the delivery
    // txn results, so cannot be used as is to run an auditable TPC-C result. For that
    // reason, delivery txns should not be done via Tuxedo.
    // The code is included for completeness.
    //m_txn->u.Delivery.exec_status_code = eDeliveryFailed;
    //return;

    // Note: If we use the delivery thread in tpcc.dll, it is not possible to get to this
    // point for delivery txns. But if we use Encina delivery server, the code is
    // needed. It is suggested using the delivery thread in tpcc.dll since it is

```

```

        // convenient and provides best performance.
        GetLocalTime(&m_txn->u.Delivery.queue_time);

    if (send_delivery(sizeof(ENC_DATA), (unsigned char *)m_txn) == TRPC_ERROR)
        m_txn->u.Delivery.exec_status_code = eDeliveryFailed;
    else
        m_txn->u.Delivery.exec_status_code = eOK;
}

void CTPCC_ENCINA::StockLevel()
{
    if (send_stock_level(sizeof(ENC_DATA), (unsigned char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn->ErrorType, m_txn->error );
}

void CTPCC_ENCINA::OrderStatus()
{
    if (send_order_status(sizeof(ENC_DATA), (unsigned char *)m_txn) == TRPC_ERROR)
        throw new CENCERR(TRPC_ERROR);

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CENCERR( m_txn->ErrorType, m_txn->error );
}

char *CENCERR::ErrorText()
{
    if (m_iErrorType == TRPC_ERROR)
    {
        sprintf( m_szErrorText, "Error: ENCINA TRPC error (see log file %s for details)",
errFile);
    }
    else
        sprintf( m_szErrorText, "Error: Class %d, error # %d", m_iErrorType, m_iError );
    return m_szErrorText;
};

```

tpcc_enc.h

```

/* FILE: TPCC_ENCINA.H
 * Microsoft TPC-C Kit Ver. 4.10.000
 * not yet audited
 *
 * PURPOSE: Header file for TPC-C Encina class implementation.
 * Copyright Microsoft, 1999
 * All Rights Reserved
 *

```

```

*/

#if !defined(_TPCC_ENCINA_H_)
#define _TPCC_ENCINA_H_

#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 CTPCC_ENCINA : public CTPCC_BASE
{
private:
    struct ENC_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_txn;

public:
    CTPCC_ENCINA();
    virtual ~CTPCC_ENCINA();

    inline PNEW_ORDER_DATA    BuffAddr_NewOrder()    { return
&m_txn->u.NewOrder; };
    inline PPAYMENT_DATA      BuffAddr_Payment()     { return
&m_txn->u.Payment; };
    inline PDELIVERY_DATA     BuffAddr_Delivery()    { return
&m_txn->u.Delivery; };
    inline PSTOCK_LEVEL_DATA  BuffAddr_StockLevel()  { return
&m_txn->u.StockLevel; };
    inline PORDER_STATUS_DATA BuffAddr_OrderStatus() { return
&m_txn->u.OrderStatus; };

    void NewOrder    ();
    void Payment     ();
    void Delivery    ();
    void StockLevel  ();
    void OrderStatus();
};

```

```

class CENCERR : public CBaseErr
{
private:
    char    m_szErrorText[64];
public:
    int     m_errno;           //
    int     m_iErrorType;    // match ErrorType in CTPCC_ENCINA
    int     m_iError;        // machine error in CTPCC_ENCINA

    // use this interface for genuine Encina errors
    CENCERR( int iErr )
    {
        m_errno = iErr; // ENCINA error
        m_iErrorType = ERR_TYPE_ENCINA;
        m_iError = 0; // only meaningful if m_errno == TPEOS
    };

    // use this interface to impersonate a non-Encina error type
    CENCERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_errno = iError; // ???
    }

    // A CENCERR class can impersonate another class, which happens if the error
    // was not actually a Tuxedo error, but was simply transmitted back via Tuxedo.
    int ErrorType()
    {
        return m_iErrorType;
    }

    int ErrorNum() {return m_errno;};
    char *ErrorText();
};

// wrapper routine for class constructor:
extern "C" __declspec(dllexport) CTPCC_ENCINA* CTPCC_ENCINA_new();
extern "C" __declspec(dllexport) CTPCC_ENCINA* CTPCC_ENCINA_post_init();

typedef CTPCC_ENCINA* (TYPE_CTPCC_ENCINA)();

#endif // !defined(_TPCC_ENCINA_H_)

```

tpcc_tux.cpp

```

/*      FILE:          TPCC_TUX.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
*/

```

```

#include <windows.h>
#include <process.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 <tmenv.h>
#include <xa.h>
#include <atmi.h>

```

```

#ifdef ICECAP
// for IceCAP profiling
#include <icapexp.h>
#endif

```

```

// 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_tux.h"                       // interface to Tuxedo libraries

```

```

static TPINIT          *tpinf;
static DWORD          TLSIsTpInitedKey;
static CRITICAL_SECTION TpCriticalSection;

```

```

BOOL WINAPI DllMain(HMODULE hModule, DWORD ul_reason_for_call, LPVOID
lpReserved)
{
    switch( ul_reason_for_call )
    {
        case DLL_PROCESS_ATTACH:

```

```

        DisableThreadLibraryCalls(hModule);

        // create thread local storage to determine Tuxedo initialization per thread.
        // it really should be possible to do this in the DLL_THREAD_ATTACH call,
but
        // Ed says he could not get it to work.
        // assumption: value init'd to 0
        TLSIsTpInitedKey = TlsAlloc();

        if ((tpinf = (TPINIT *)tpalloc("TPINIT", NULL, sizeof(TPINIT))) ==
NULL)
        {
            // int TpRc = tperno;
            return FALSE;
        }
        tpinf->flags |= TPMULTICONTEXTS;

        InitializeCriticalSection(&TpCriticalSection);
        break;

    case DLL_PROCESS_DETACH:
        TlsFree(TLSIsTpInitedKey);
        DeleteCriticalSection(&TpCriticalSection);
        break;

    default:
        /* nothing */;
}
return TRUE;
}

static void ThrTpInit()
{
    static int num_tpinit=0;
    int iRc, TpRc;

    // has this thread been initialized? check thread local storage
    if(!TlsGetValue(TLSIsTpInitedKey))
    {
        EnterCriticalSection(&TpCriticalSection);
        itoa(++num_tpinit, tpinf->cltname, 10);

        iRc = tpinit(tpinf);
        TpRc = tperno;
        LeaveCriticalSection(&TpCriticalSection);

        if (iRc < 0)
            throw new CTUXERR( tperno );

        int value = 1;
        TlsSetValue(TLSIsTpInitedKey, &value);
    }
}

```



```

// wrapper routine for class constructor
__declspec(dllexport) CTPCC_TUXEDO* CTPCC_TUXEDO_new()
{
    return new CTPCC_TUXEDO();
}

CTPCC_TUXEDO::CTPCC_TUXEDO()
{
    //      Add initialization of Tuxedo Structures
    m_txn = (TUX_DATA *)tpalloc("CARRAY", NULL, sizeof(TUX_DATA));
    if (m_txn == NULL)
        throw new CTUXERR( tperno );
}

CTPCC_TUXEDO::~~CTPCC_TUXEDO()
{
    // free the data structure allocated with tpalloc
    tpfree((char *)m_txn);
}

void CTPCC_TUXEDO::NewOrder()
{
    long    ilen, *olen;

    ThrTpInit();

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if (tpcall("NEWORDER", (char *)m_txn, ilen, (char **)&m_txn, (long *)olen, TPSIGRSTRT)
== -1)
        throw new CTUXERR( tperno );

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CTUXERR( m_txn->ErrorType, m_txn->error );
}

void CTPCC_TUXEDO::Payment()
{
    long    ilen, *olen;

    ThrTpInit();

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if (tpcall("PAYMENT", (char *)m_txn, ilen, (char **)&m_txn, (long *)olen, TPSIGRSTRT)
== -1)
        throw new CTUXERR( tperno );

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CTUXERR( m_txn->ErrorType, m_txn->error );
}

```

```

}

void CTPCC_TUXEDO::Delivery()
{
    int      iRc;
    long    ilen, *olen;

    // Note: Delivery txn code in the tuxedo server does not implement logging of the delivery
    //       txn results, so cannot be used as is to run an auditable TPC-C result. For that
    //       reason, delivery txns should not be done via tuxedo.
    //       The code is included for completeness.
    m_txn->u.Delivery.exec_status_code = eDeliveryFailed;
    return;

    // normal path...

    ThrTpInit();

    GetLocalTime(&m_txn->u.Delivery.queue_time);

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if ((iRc = tpcall("DELIVERY", (char *)m_txn, ilen, TPNOREPLY)) == -1)
    {
        int TpRc = tperrno;
        m_txn->u.Delivery.exec_status_code = eDeliveryFailed;
    }
    else
        m_txn->u.Delivery.exec_status_code = eOK;
}

void CTPCC_TUXEDO::StockLevel()
{
    long    ilen, *olen;

    ThrTpInit();

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if (tpcall("STOCKLEVEL", (char *)m_txn, ilen, (char **)&m_txn, (long *)olen,
    TPSIGRSTRT) == -1)
        throw new CTUXERR( tperrno );

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CTUXERR( m_txn->ErrorType, m_txn->error );
}

void CTPCC_TUXEDO::OrderStatus()
{
    long    ilen, *olen;

```

```

    ThrTpInit();

    ilen = sizeof(TUX_DATA);
    olen = &ilen;

    if (tpcall("ORDERSTATUS", (char *)m_txn, ilen, (char **)&m_txn, (long *)olen,
TPSIGRSTRT) == -1)
        throw new CTUXERR( tperrno );

    if ( m_txn->ErrorType != ERR_SUCCESS )
        throw new CTUXERR( m_txn->ErrorType, m_txn->error );
}

char *CTUXERR::ErrorText()
{
    if (m_iErrorType == 0)
    {
        if (m_erno == TPEOS)
            sprintf( m_szErrorText, "Error: TUXEDO error # %d, OS error # %d",
m_erno, m_iError );
        else
            sprintf( m_szErrorText, "Error: TUXEDO error # %d", m_erno );
    }
    else
        sprintf( m_szErrorText, "Error: Class %d, error # %d", m_iErrorType, m_iError );
    return m_szErrorText;
};

```

tpcc_tux.h

```

/*      FILE:          TPCC_TUX.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 Tuxedo class implementation.
*
*      Change history:
*                      4.20.000 - updated rev number to match kit
*/

#pragma once

// need to declare functions for import, unless define has already been created
// by the DLL's .cpp module for export.
#ifdef DllDecl
#define DllDecl __declspec( dllimport )
#endif

```

```

class DllDecl CTPCC_TUXEDO : public CTPCC_BASE
{
    private:
        struct TUX_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_txn;

    public:
        CTPCC_TUXEDO();
        ~CTPCC_TUXEDO(void);

        inline PNEW_ORDER_DATA      BuffAddr_NewOrder()      { return
&m_txn->u.NewOrder; };
        inline PPAYMENT_DATA        BuffAddr_Payment()       { return
&m_txn->u.Payment; };
        inline PDELIVERY_DATA       BuffAddr_Delivery()      { return
&m_txn->u.Delivery; };
        inline PSTOCK_LEVEL_DATA    BuffAddr_StockLevel()    { return
&m_txn->u.StockLevel; };
        inline PORDER_STATUS_DATA   BuffAddr_OrderStatus()   { return
&m_txn->u.OrderStatus; };

        void NewOrder      ();
        void Payment       ();
        void Delivery      ();
        void StockLevel    ();
        void OrderStatus();

};

class CTUXERR : public CBaseErr
{
    private:
        // TODO: should use the sz_Msg field of the base class instead
        char m_szErrorText[64];

    public:
        // use this interface for genuine Tuxedo errors
        CTUXERR( int iErr )
        {

```

```

        m_errno = iErr;
        m_iErrorType = 0;
        m_iError = GetLastError();    // only meaningful if m_errno == TPEOS
    };

    // use this interface to impersonate a non-Tuxedo error type
    CTUXERR( int iErrorType, int iError )
    {
        m_iErrorType = iErrorType;
        m_iError = iError;
        m_errno = 0;
    }

    int          m_errno;
    int          m_iErrorType;
    int          m_iError;

    // A CTUXERR class can impersonate another class, which happens if the error
    // was not actually a Tuxedo error, but was simply transmitted back via Tuxedo.
    int ErrorType()
    {
        if (m_iErrorType == 0)
            return ERR_TYPE_TUXEDO;
        else
            return m_iErrorType;
    }

    int ErrorNum() {return m_errno;};
    char *ErrorText();
};

// wrapper routine for class constructor
extern "C" __declspec(dllexport) CTPCC_TUXEDO* CTPCC_TUXEDO_new();

typedef CTPCC_TUXEDO* (TYPE_CTPCC_TUXEDO)();

```

Methods.h

```

/*      FILE:          METHODS.H
 *
 *                  Microsoft TPC-C Kit Ver. 4.20.000
 *                  Copyright Microsoft, 1999
 *
 *      All Rights Reserved
 *
 *
 *                  not yet audited
 *
 *      PURPOSE:      Header file for COM components.
 *
 *      Change history:
 *          4.20.000 - first version
 */

```

```

enum COMPONENT_ERROR
{
    ERR_MISSING_REGISTRY_ENTRIES = 1,
    ERR_LOADDLL_FAILED,
    ERR_GETPROCADDR_FAILED,
    ERR_UNKNOWN_DB_PROTOCOL
};

class CCOMPONENT_ERR : public CBaseErr
{
public:
    CCOMPONENT_ERR(COMPONENT_ERROR Err)
    {
        m_Error = Err;
        m_szTextDetail = NULL;
        m_SystemErr = 0;
        m_szErrorText = NULL;
    };

    CCOMPONENT_ERR(COMPONENT_ERROR Err, char *szTextDetail, DWORD
dwSystemErr)
    {
        m_Error = Err;
        m_szTextDetail = new char[strlen(szTextDetail)+1];
        strcpy( m_szTextDetail, szTextDetail );
        m_SystemErr = dwSystemErr;
        m_szErrorText = NULL;
    };

    ~CCOMPONENT_ERR()
    {
        if (m_szTextDetail != NULL)
            delete [] m_szTextDetail;
        if (m_szErrorText != NULL)
            delete [] m_szErrorText;
    };

    COMPONENT_ERROR m_Error;
    char             *m_szTextDetail;
    char             *m_szErrorText;
    DWORD            m_SystemErr;

    int ErrorType() {return ERR_TYPE_COMPONENT;};
    char *ErrorTypeStr() { return "COMPONENT"; }
    int ErrorNum() {return m_Error;};
    char *ErrorText();
};

static void WriteMessageToEventLog(LPTSTR lpszMsg);

```

```

////////////////////////////////////

```

```

// CTPCC_Common
class CTPCC_Common :
    public ITPCC,
    public IObjectControl,
    public IObjectConstruct,
    public CComObjectRootEx<CComSingleThreadModel>
{
public:
BEGIN_COM_MAP(CTPCC_Common)
    COM_INTERFACE_ENTRY(ITPCC)
    COM_INTERFACE_ENTRY(IObjectControl)
    COM_INTERFACE_ENTRY(IObjectConstruct)
END_COM_MAP()

    CTPCC_Common();
    ~CTPCC_Common();

// ITPCC
public:
    HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out);
    HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out);
    HRESULT __stdcall Delivery(         VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;};
    HRESULT __stdcall StockLevel(       VARIANT txn_in, VARIANT* txn_out);
    HRESULT __stdcall OrderStatus(      VARIANT txn_in, VARIANT* txn_out);

    HRESULT __stdcall CallSetComplete();

// IObjectControl
    STDMETHODCALLTYPE CanBePooled() { return m_bCanBePooled; }
    STDMETHODCALLTYPE Activate() { return S_OK; } // we don't support COM Services
transactions (no enlistment)
    STDMETHODCALLTYPE Deactivate() { /* nothing to do */ }

// IObjectConstruct
    STDMETHODCALLTYPE Construct(IDispatch * pUnk);

    // helper methods
private:
    BOOL            m_bCanBePooled;
    CTPCC_BASE     *m_pTxn;

    struct COM_DATA
    {
        int retval;
        int error;
        union
        {
            NEW_ORDER_DATA           NewOrder;
            PAYMENT_DATA              Payment;
            DELIVERY_DATA             Delivery;
            STOCK_LEVEL_DATA StockLevel;
            ORDER_STATUS_DATA        OrderStatus;
        }
    }

```

```

        } u;
};

};

////////////////////////////////////
// CTPCC
class CTPCC :
    public CTPCC_Common,
    public CComCoClass<CTPCC, &CLSID_TPCC>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_TPCC)

BEGIN_COM_MAP(CTPCC)
    //COM_INTERFACE_ENTRY2(IUnknown,
    CComObjectRootEx<CComSingleThreadModel>)
    COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

};

////////////////////////////////////
// CNewOrder
class CNewOrder :
    public CTPCC_Common,
    public CComCoClass<CNewOrder, &CLSID_NewOrder>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_NEWORDER)

BEGIN_COM_MAP(CNewOrder)
//    COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
    COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
    COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
//    HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel(  VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
};

////////////////////////////////////

```



```

// COrderStatus
class COrderStatus :
    public CTPCC_Common,
    public CComCoClass<COrderStatus, &CLSID_OrderStatus>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_ORDERSTATUS)

BEGIN_COM_MAP(COrderStatus)
//     COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
//     COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//     COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel(  VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
//     HRESULT __stdcall OrderStatus(VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
};

/////////////////////////////////////////////////////////////////
// CPayment
class CPayment :
    public CTPCC_Common,
    public CComCoClass<CPayment, &CLSID_Payment>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_PAYMENT)

BEGIN_COM_MAP(CPayment)
//     COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
//     COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//     COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
//     HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall StockLevel(  VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
    HRESULT __stdcall OrderStatus(VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
};

```

```

////////////////////////////////////
// CStockLevel
class CStockLevel :
    public CTPCC_Common,
    public CComCoClass<CStockLevel, &CLSID_StockLevel>
{
public:
DECLARE_REGISTRY_RESOURCEID(IDR_STOCKLEVEL)

BEGIN_COM_MAP(CStockLevel)
//     COM_INTERFACE_ENTRY2(IUnknown, CComObjectRootEx)
//     COM_INTERFACE_ENTRY2(IUnknown, ITPCC)
//     COM_INTERFACE_ENTRY_CHAIN(CTPCC_Common)
END_COM_MAP()

// ITPCC
public:
    HRESULT __stdcall NewOrder(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
    HRESULT __stdcall Payment(          VARIANT txn_in, VARIANT* txn_out) {return
E_NOTIMPL;}
//     HRESULT __stdcall StockLevel( VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
    HRESULT __stdcall OrderStatus( VARIANT   txn_in,   VARIANT*   txn_out)   {return
E_NOTIMPL;}
};

```

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
#ifdef APSTUDIO_READONLY_SYMBOLS
#define _APS_NEXT_RESOURCE_VALUE 202
#define _APS_NEXT_COMMAND_VALUE 32768
#define _APS_NEXT_CONTROL_VALUE 201
#define _APS_NEXT_SYMED_VALUE 106
#endif
#endif

```

tpcc_com_all.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;

// Critical section to synchronize connection open and close.
//
CRITICAL_SECTION hConnectCriticalSection;

////////////////////////////////////
// DLL Entry Point

extern "C"
BOOL WINAPI DllMain(HINSTANCE hInstance, DWORD dwReason, LPVOID /*lpReserved*/)
{
    char szDllName[128];

    try
    {
        if (dwReason == DLL_PROCESS_ATTACH)
        {
            _Module.Init(ObjectMap, hInstance);
            DisableThreadLibraryCalls(hInstance);

            DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
            GetComputerName(szMyComputerName, &dwSize);
            szMyComputerName[dwSize] = 0;

            if ( ReadTPCCRegistrySettings( &Reg ) )
                throw new
CCOMPONENT_ERR( ERR_MISSING_REGISTRY_ENTRIES );

            if (Reg.eDB_Protocol == DBLIB)
            {
                strcpy( szDllName, Reg.szPath );
            }
        }
    }
}

```

```

        strcat( szDllName, "tpcc_dblib.dll");
        hLibInstanceDb = LoadLibrary( szDllName );
        if (hLibInstanceDb == NULL)
            throw new
CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

        // get function pointer to wrapper for class constructor
        pCTPCC_DBLIB_new = (TYPE_CTPCC_DBLIB*)
GetProcAddress(hLibInstanceDb,"CTPCC_DBLIB_new");
        if (pCTPCC_DBLIB_new == NULL)
            throw new
CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
    }
    else if (Reg.eDB_Protocol == ODBC)
    {
        strcpy( szDllName, Reg.szPath );
        strcat( szDllName, "tpcc_odbc.dll");
        hLibInstanceDb = LoadLibrary( szDllName );
        if (hLibInstanceDb == NULL)
            throw new
CCOMPONENT_ERR( ERR_LOADDLL_FAILED, szDllName, GetLastError() );

        // get function pointer to wrapper for class constructor
        pCTPCC_ODBC_new = (TYPE_CTPCC_ODBC*)
GetProcAddress(hLibInstanceDb,"CTPCC_ODBC_new");
        if (pCTPCC_ODBC_new == NULL)
            throw new
CCOMPONENT_ERR( ERR_GETPROCADDR_FAILED, szDllName, GetLastError() );
    }
    else
        throw new
CCOMPONENT_ERR( ERR_UNKNOWN_DB_PROTOCOL );

        if (Reg.dwConnectDelay > 0)
        {
            InitializeCriticalSection(&hConnectCriticalSection);
        }

    }
    else if (dwReason == DLL_PROCESS_DETACH)
        _Module.Term();

}
catch (CBaseErr *e)
{
    TCHAR szMsg[256];

    _sntprintf(szMsg, sizeof(szMsg), "%s error, code %d: %s",
        e->ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
    WriteMessageToEventLog( szMsg );
    delete e;
    return FALSE;
}

```

```

        catch (...)
        {
            WriteMessageToEventLog(TEXT("Unhandled exception in object DllMain"));
            return FALSE;
        }

        return TRUE;    // OK
    }

    ////////////////////////////////////////////////////////////////////
    // Used to determine whether the DLL can be unloaded by OLE

    STDAPI DllCanUnloadNow(void)
    {
        return (_Module.GetLockCount()==0) ? S_OK : S_FALSE;
    }

    ////////////////////////////////////////////////////////////////////
    // Returns a class factory to create an object of the requested type

    STDAPI DllGetClassObject(REFCLSID reclsid, REFIID riid, LPVOID* ppv)
    {
        return _Module.GetClassObject(reclsid, 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[] =
    {
        registry." { ERR_MISSING_REGISTRY_ENTRIES, "Required entries missing from
        { 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)
        sprintf( szTmp+strlen(szTmp), " Error=%d", m_SystemErr );

    m_szErrorText = new char[strlen(szTmp)+1];
    strcpy( m_szErrorText, szTmp );
    return m_szErrorText;
}

```

```

CTPCC_Common::CTPCC_Common()
{
    m_pTxn = NULL;
    m_bCanBePooled = TRUE;
}

```

```

CTPCC_Common::~~CTPCC_Common()
{
    // Pace connection close for VIA.
    //
    if (Reg.dwConnectDelay > 0)
    {
        EnterCriticalSection(&hConnectCriticalSection);

        Sleep(Reg.dwConnectDelay);
    }

    if (m_pTxn)
    {
        delete m_pTxn;
    }

    if (Reg.dwConnectDelay > 0)
    {
        LeaveCriticalSection(&hConnectCriticalSection);
    }
}

```



```

HRESULT CTPCC_Common::CallSetComplete()
{
    IObjectContext* pObjectContext = NULL;

    // get our object context
    HRESULT hr = CoGetObjectContext( IID_IObjectContext, (void **)&pObjectContext );
    pObjectContext->SetComplete();
    ReleaseInterface(pObjectContext);
    return hr;
}

//
// called by the ctor activator
//
STDMETHODIMP CTPCC_Common::Construct(IDispatch * pUnk)
{
    // Code to access construction string, if needed later...
    // if (!pUnk)
    //     return E_UNEXPECTED;
    // IObjectConstructString * pString = NULL;
    // HRESULT hr = pUnk->QueryInterface(IID_IObjectConstructString, (void
    **)&pString);
    // pString->Release();

    try
    {
        // Pace connection creation for VIA.
        //
        if (Reg.dwConnectDelay > 0)
        {
            EnterCriticalSection(&hConnectCriticalSection);

            Sleep(Reg.dwConnectDelay);
        }

        if (Reg.eDB_Protocol == ODBC)
            m_pTxn = pCTPCC_ODBC_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword,
            szMyComputerName, Reg.szDbName,
            Reg.szSPPrefix,
Reg.bCallNoDuplicatesNewOrder );
        else if (Reg.eDB_Protocol == DBLIB)
            m_pTxn = pCTPCC_DBLIB_new( Reg.szDbServer, Reg.szDbUser,
Reg.szDbPassword, szMyComputerName, Reg.szDbName );

        if (Reg.dwConnectDelay > 0)
        {
            LeaveCriticalSection(&hConnectCriticalSection);
        }
    }
    catch (CBaseErr *e)
    {
    }
}

```

```

        TCHAR szMsg[256];

        _sntprintf(szMsg, sizeof(szMsg), "%s error in CTPCC_Common::Construct, code
%d: %s",
                e->ErrorTypeStr(), e->ErrorNum(), e->ErrorText());
        WriteMessageToEventLog( szMsg );
        delete e;
        return E_FAIL;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("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.h

```
#pragma warning( disable: 4049 ) /* more than 64k source lines */
```

```
/* this ALWAYS GENERATED file contains the definitions for the interfaces */
```

```
/* File created by MIDL compiler version 6.00.0347 */
```

```
/* at Fri Apr 15 14:48:53 2005
```

```
*/
```

```
/* Compiler settings for .\src\tpcc_com_all.idl:
```

```
    Oicf, W1, Zp8, env=Win32 (32b run)
```

```
    protocol : dce , ms_ext, c_ext
```

```
    error checks: allocation ref bounds_check enum stub_data
```

```
    VC __declspec() decoration level:
```

```
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
```

```
        DECLSPEC_UUID(), MIDL_INTERFACE()
```

```
*/
```

```
//@@MIDL_FILE_HEADING( )
```

```
/* verify that the <rpcndr.h> version is high enough to compile this file*/
```

```
#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__
```

```
#if defined(_MSC_VER) && (_MSC_VER >= 1020)
```

```
#pragma once
```

```
#endif
```

```
/* Forward Declarations */
```

```
#ifndef __TPCC_FWD_DEFINED__
```

```

#define __TPCC_FWD_DEFINED__

#ifdef __cplusplus
typedef class TPCC TPCC;
#else
typedef struct TPCC TPCC;
#endif /* __cplusplus */

#endif /* __TPCC_FWD_DEFINED__ */

#ifndef __NewOrder_FWD_DEFINED__
#define __NewOrder_FWD_DEFINED__

#ifdef __cplusplus
typedef class NewOrder NewOrder;
#else
typedef struct NewOrder NewOrder;
#endif /* __cplusplus */

#endif /* __NewOrder_FWD_DEFINED__ */

#ifndef __OrderStatus_FWD_DEFINED__
#define __OrderStatus_FWD_DEFINED__

#ifdef __cplusplus
typedef class OrderStatus OrderStatus;
#else
typedef struct OrderStatus OrderStatus;
#endif /* __cplusplus */

#endif /* __OrderStatus_FWD_DEFINED__ */

#ifndef __Payment_FWD_DEFINED__
#define __Payment_FWD_DEFINED__

#ifdef __cplusplus
typedef class Payment Payment;
#else
typedef struct Payment Payment;
#endif /* __cplusplus */

#endif /* __Payment_FWD_DEFINED__ */

#ifndef __StockLevel_FWD_DEFINED__
#define __StockLevel_FWD_DEFINED__

#ifdef __cplusplus
typedef class StockLevel StockLevel;
#else

```

```

typedef struct StockLevel StockLevel;
#endif /* __cplusplus */

#endif /* __StockLevel_FWD_DEFINED__ */

/* header files for imported files */
#include "oidl.h"
#include "ocidl.h"
#include "tpcc_com_ps.h"

#ifdef __cplusplus
extern "C"{
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_all_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_all_0000_v0_0_s_ifspec;

#ifdef __TPCCLib_LIBRARY_DEFINED__
#define __TPCCLib_LIBRARY_DEFINED__

/* library TPCCLib */
/* [helpstring][version][uuid] */

EXTERN_C const IID LIBID_TPCCLib;

EXTERN_C const CLSID CLSID_TPCC;

#ifdef __cplusplus
class DECLSPEC_UUID("122A3128-2520-11D3-BA71-00C04FBFE08B")
TPCC;
#endif

EXTERN_C const CLSID CLSID_NewOrder;

#ifdef __cplusplus

```



```

class DECLSPEC_UUID("975BAABF-84A7-11D2-BA47-00C04FBFE08B")
NewOrder;
#endif

EXTERN_C const CLSID CLSID_OrderStatus;

#ifdef __cplusplus

class DECLSPEC_UUID("266836AD-A50D-11D2-BA4E-00C04FBFE08B")
OrderStatus;
#endif

EXTERN_C const CLSID CLSID_Payment;

#ifdef __cplusplus

class DECLSPEC_UUID("CD02F7EF-A4FA-11D2-BA4E-00C04FBFE08B")
Payment;
#endif

EXTERN_C const CLSID CLSID_StockLevel;

#ifdef __cplusplus

class DECLSPEC_UUID("2668369E-A50D-11D2-BA4E-00C04FBFE08B")
StockLevel;
#endif
#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

#ifdef _MAC
////////////////////////////////////
//
// Version
//

VS_VERSION_INFO VERSIONINFO
FILEVERSION 1,0,0,1
PRODUCTVERSION 1,0,0,1
FILEFLAGSMASK 0x3fL
#ifdef _DEBUG
    FILEFLAGS 0x1L
#else
    FILEFLAGS 0x0L
#endif
FILEOS 0x4L
FILETYPE 0x2L
FILESUBTYPE 0x0L
BEGIN
    BLOCK "StringFileInfo"
    BEGIN
        BLOCK "040904B0"
        BEGIN
            VALUE "CompanyName", "\0"
            VALUE "FileDescription", "tpcc_com_all Module\0"
            VALUE "FileVersion", "1, 0, 0, 1\0"
            VALUE "InternalName", "TPCCNEWORDER\0"
            VALUE "LegalCopyright", "Copyright 1997\0"
            VALUE "OriginalFilename", "tpcc_com_all.DLL\0"
            VALUE "ProductName", "tpcc_com_all Module\0"
        END
    END
END

```

```

        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 6.00.0347 */
/* at Fri Apr 15 14:48:53 2005
*/
/* Compiler settings for .\src\tpcc_com_all.idl:
Oicf, W1, Zp8, env=Win32 (32b run)
protocol : dce , ms_ext, c_ext
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
    __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)

#ifdef __cplusplus
```

```

extern "C"{
#endif

#include <rpc.h>
#include <rpcndr.h>

#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_AMD64)*/
```

```
#pragma warning( disable: 4049 ) /* more than 64k source lines */
```

```
/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */
```

```
/* link this file in with the server and any clients */
```

```
/* File created by MIDL compiler version 6.00.0347 */
```

```
/* at Fri Apr 15 14:48:53 2005
```

```
*/
```

```
/* Compiler settings for .\src\tpcc_com_all.idl:
```

```
Oicf, W1, Zp8, env=Win64 (32b run,appending)
```

```
protocol : dce , ms_ext, c_ext, robust
```

```
error checks: allocation ref bounds_check enum stub_data
```

```
VC __declspec() decoration level:
```

```
__declspec(uuid()), __declspec(selectany), __declspec(novtable)
```

```
DECLSPEC_UUID(), MIDL_INTERFACE()
```

```
*/
```

```
//@@MIDL_FILE_HEADING( )
```

```
#if defined(_M_IA64) || defined(_M_AMD64)
```

```
#ifdef __cplusplus
```

```
extern "C" {
```

```
#endif
```



```

#include <rpc.h>
#include <rpendr.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_AMD64) */
```

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.h

```
#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */

/* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
   Oicf, W1, Zp8, env=Win32 (32b run)
   protocol : dce , ms_ext, c_ext
   error checks: allocation ref bounds_check enum stub_data
   VC __declspec() decoration level:
       __declspec(uuid()), __declspec(selectany), __declspec(novtable)
   DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifndef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

#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__

#ifdef _MSC_VER && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifndef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
```

```

typedef interface ITPCC ITPCC;
#endif /* __ITPCC_FWD_DEFINED__ */

/* header files for imported files */
#include "oidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C" {
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifdef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

EXTERN_C const IID IID_ITPCC;

#ifdef __cplusplus && !defined(CINTERFACE)

MIDL_INTERFACE("FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B")
ITPCC : public IUnknown
{
public:
    virtual HRESULT __stdcall NewOrder(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Payment(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Delivery(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall StockLevel(
        /* [in] */ VARIANT txn_in,

```

```

        /* [out] */ VARIANT *txn_out) = 0;

virtual HRESULT __stdcall OrderStatus(
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out) = 0;

virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else    /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
        ITPCC * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef )(
        ITPCC * This);

    ULONG ( STDMETHODCALLTYPE *Release )(
        ITPCC * This);

    HRESULT ( __stdcall *NewOrder )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *Payment )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *Delivery )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *StockLevel )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *OrderStatus )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

    HRESULT ( __stdcall *CallSetComplete )(

```

```

        ITPCC * This);

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl *lpVtbl;
};

#ifdef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This) \
    (This)->lpVtbl -> Release(This)

#define ITPCC_NewOrder(This,txn_in,txn_out) \
    (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out) \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out) \
    (This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out) \
    (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
    (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This) \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */

#endif /* C style interface */

HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,

```

```

/* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

```



```

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

```

```

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

#endif /* __ITPCC_INTERFACE_DEFINED__ */

```

```

/* Additional Prototypes for ALL interfaces */

```

```

unsigned long __RPC_USER VARIANT_UserSize( unsigned long *, unsigned long
    ,
    VARIANT * );
unsigned char * __RPC_USER VARIANT_UserMarshal( unsigned long *, unsigned char *,
    VARIANT * );
unsigned char * __RPC_USER VARIANT_UserUnmarshal(unsigned long *, unsigned char *,
    VARIANT * );
void __RPC_USER VARIANT_UserFree( unsigned long *, VARIANT * );

```

```

/* end of Additional Prototypes */

```

```

#ifdef __cplusplus
}
#endif

```

```

#endif

```

tpcc_com_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'
            }
        }
    }
}

```

dlldata.c

```

/*****

```

DllData file -- generated by MIDL compiler

DO NOT ALTER THIS FILE

This file is regenerated by MIDL on every IDL file compile.

To completely reconstruct this file, delete it and rerun MIDL on all the IDL files in this DLL, specifying this file for the /dlldata command line option

```

*****/

```

```

#include <rpcproxy.h>

```

```

#ifdef __cplusplus
extern "C" {
#endif

```

```

EXTERN_PROXY_FILE( tpcc_com_ps )

```

```

PROXYFILE_LIST_START

```

```

/* Start of list */

```

```

REFERENCE_PROXY_FILE( tpcc_com_ps ),

```

```

/* End of list */

```

```

PROXYFILE_LIST_END

```

```

DLLDATA_ROUTINES( aProxyFileList, GET_DLL_CLSID )

```

```

#ifdef __cplusplus
} /*extern "C" */
#endif

/* end of generated dlldata file */

```

tpcc_com_ps.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.h

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the definitions for the interfaces */

/* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

/* verify that the <rpcndr.h> version is high enough to compile this file*/
#ifdef __REQUIRED_RPCNDR_H_VERSION__
#define __REQUIRED_RPCNDR_H_VERSION__ 440
#endif

#include "rpc.h"
#include "rpcndr.h"

```

```

#ifdef __RPCNDR_H_VERSION__
#error this stub requires an updated version of <rpcndr.h>
#endif // __RPCNDR_H_VERSION__

#ifdef COM_NO_WINDOWS_H
#include "windows.h"
#include "ole2.h"
#endif /*COM_NO_WINDOWS_H*/

#ifdef __tpcc_com_ps_h__
#define __tpcc_com_ps_h__

#if defined(_MSC_VER) && (_MSC_VER >= 1020)
#pragma once
#endif

/* Forward Declarations */

#ifdef __ITPCC_FWD_DEFINED__
#define __ITPCC_FWD_DEFINED__
typedef interface ITPCC ITPCC;
#endif /* __ITPCC_FWD_DEFINED__ */

/* header files for imported files */
#include "oaidl.h"
#include "ocidl.h"

#ifdef __cplusplus
extern "C" {
#endif

void * __RPC_USER MIDL_user_allocate(size_t);
void __RPC_USER MIDL_user_free( void * );

/* interface __MIDL_itf_tpcc_com_ps_0000 */
/* [local] */

extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_c_ifspec;
extern RPC_IF_HANDLE __MIDL_itf_tpcc_com_ps_0000_v0_0_s_ifspec;

#ifdef __ITPCC_INTERFACE_DEFINED__
#define __ITPCC_INTERFACE_DEFINED__

/* interface ITPCC */
/* [unique][helpstring][uuid][oleautomation][object] */

EXTERN_C const IID IID_ITPCC;

```

```

#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 *txn_out) = 0;

    virtual HRESULT __stdcall Payment(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall Delivery(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall StockLevel(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall OrderStatus(
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out) = 0;

    virtual HRESULT __stdcall CallSetComplete( void) = 0;

};

#else /* C style interface */

typedef struct ITPCCVtbl
{
    BEGIN_INTERFACE

    HRESULT ( STDMETHODCALLTYPE *QueryInterface )(
        ITPCC * This,
        /* [in] */ REFIID riid,
        /* [iid_is][out] */ void **ppvObject);

    ULONG ( STDMETHODCALLTYPE *AddRef )(
        ITPCC * This);

    ULONG ( STDMETHODCALLTYPE *Release )(
        ITPCC * This);

    HRESULT ( __stdcall *NewOrder )(
        ITPCC * This,
        /* [in] */ VARIANT txn_in,
        /* [out] */ VARIANT *txn_out);

```

```

HRESULT ( __stdcall *Payment )(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

HRESULT ( __stdcall *Delivery )(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

HRESULT ( __stdcall *StockLevel )(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

HRESULT ( __stdcall *OrderStatus )(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

HRESULT ( __stdcall *CallSetComplete )(
    ITPCC * This);

    END_INTERFACE
} ITPCCVtbl;

interface ITPCC
{
    CONST_VTBL struct ITPCCVtbl *lpVtbl;
};

#ifdef COBJMACROS

#define ITPCC_QueryInterface(This,riid,ppvObject) \
    (This)->lpVtbl -> QueryInterface(This,riid,ppvObject)

#define ITPCC_AddRef(This) \
    (This)->lpVtbl -> AddRef(This)

#define ITPCC_Release(This) \
    (This)->lpVtbl -> Release(This)

#define ITPCC_NewOrder(This,txn_in,txn_out) \
    (This)->lpVtbl -> NewOrder(This,txn_in,txn_out)

#define ITPCC_Payment(This,txn_in,txn_out) \
    (This)->lpVtbl -> Payment(This,txn_in,txn_out)

#define ITPCC_Delivery(This,txn_in,txn_out) \

```

```

(This)->lpVtbl -> Delivery(This,txn_in,txn_out)

#define ITPCC_StockLevel(This,txn_in,txn_out) \
    (This)->lpVtbl -> StockLevel(This,txn_in,txn_out)

#define ITPCC_OrderStatus(This,txn_in,txn_out) \
    (This)->lpVtbl -> OrderStatus(This,txn_in,txn_out)

#define ITPCC_CallSetComplete(This) \
    (This)->lpVtbl -> CallSetComplete(This)

#endif /* COBJMACROS */

```

```

#endif /* C style interface */

```

```

HRESULT __stdcall ITPCC_NewOrder_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

```

```

void __RPC_STUB ITPCC_NewOrder_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

HRESULT __stdcall ITPCC_Payment_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

```

```

void __RPC_STUB ITPCC_Payment_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,
    PRPC_MESSAGE pRpcMessage,
    DWORD *_pdwStubPhase);

```

```

HRESULT __stdcall ITPCC_Delivery_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

```

```

void __RPC_STUB ITPCC_Delivery_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer *pRpcChannelBuffer,

```

```

PRPC_MESSAGE _pRpcMessage,
DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_StockLevel_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_StockLevel_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer * _pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_OrderStatus_Proxy(
    ITPCC * This,
    /* [in] */ VARIANT txn_in,
    /* [out] */ VARIANT *txn_out);

void __RPC_STUB ITPCC_OrderStatus_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer * _pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

HRESULT __stdcall ITPCC_CallSetComplete_Proxy(
    ITPCC * This);

void __RPC_STUB ITPCC_CallSetComplete_Stub(
    IRpcStubBuffer *This,
    IRpcChannelBuffer * _pRpcChannelBuffer,
    PRPC_MESSAGE _pRpcMessage,
    DWORD *_pdwStubPhase);

#endif /* __ITPCC_INTERFACE_DEFINED__ */

/* Additional Prototypes for ALL interfaces */

unsigned long         __RPC_USER VARIANT_UserSize( unsigned long *, unsigned long
VARIANT * );
unsigned char * __RPC_USER VARIANT_UserMarshal( unsigned long *, unsigned char *,
VARIANT * );
unsigned char * __RPC_USER VARIANT_UserUnmarshal(unsigned long *, unsigned char *,
VARIANT * );

```



```
void          __RPC_USER VARIANT_UserFree( unsigned long *, VARIANT * );
```

```
/* end of Additional Prototypes */
```

```
#ifdef __cplusplus  
}  
#endif
```

```
#endif
```

tpcc_com_ps.idl

```
/*      FILE:          ITPCC.IDL  
 *              Microsoft TPC-C Kit Ver. 4.20.000  
 *              Copyright Microsoft, 1999  
 *      All Rights Reserved  
 *              not yet audited  
 *      PURPOSE:      Defines the interface used by TPCC. This interface can be implemented by  
 C++ components.  
 *      Change history:  
 *          4.20.000 - first version  
 */
```

```
// Forward declare all types defined
```

```
interface ITPCC;  
import "oidl.idl";  
import "ocidl.idl";
```

```
[  
    object,  
    oleautomation,  
    uuid(FEEE6AA2-84B1-11d2-BA47-00C04FBFE08B),  
    helpstring("ITPCC Interface"),  
    pointer_default(unique)  
]  
interface ITPCC : IUnknown  
{
```

```
    HRESULT _stdcall NewOrder
```

```
(  
    [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 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext
    error checks: allocation ref bounds_check enum stub_data
    VC __declspec() decoration level:
        __declspec(uuid()), __declspec(selectany), __declspec(novtable)
        DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@@MIDL_FILE_HEADING( )

```

```

#if !defined(_M_IA64) && !defined(_M_AMD64)

#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_AMD64)*/

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the IIDs and CLSIDs */

/* link this file in with the server and any clients */

/* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
Oicf, W1, Zp8, env=Win64 (32b run,appending)
protocol : dce , ms_ext, c_ext, robust
error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
    __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#ifdef _M_IA64 || defined(_M_AMD64)

#ifdef __cplusplus
extern "C" {
#endif

#include <rpc.h>
#include <rpcndr.h>

#ifdef _MIDL_USE_GUIDDEF_

#ifndef INITGUID
#define INITGUID
#include <guiddef.h>
#undef INITGUID
#else
#include <guiddef.h>
#endif

#define MIDL_DEFINE_GUID(type,name,l,w1,w2,b1,b2,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_AMD64)*/

```

tpcc_com_ps_p.c

```

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
    Oicf, W1, Zp8, env=Win32 (32b run)
    protocol : dce , ms_ext, c_ext

```

```

error checks: allocation ref bounds_check enum stub_data
VC __declspec() decoration level:
    __declspec(uuid()), __declspec(selectany), __declspec(novtable)
    DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if !defined(_M_IA64) && !defined(_M_AMD64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 440
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1023
#define PROC_FORMAT_STRING_SIZE 193
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short    Pad;
    unsigned char    Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short    Pad;
    unsigned char    Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

```

```

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;

extern          const          USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined( __RPC_WIN32__ )
#error Invalid build platform for this stub.
#endif

#if !(TARGET_IS_NT40_OR_LATER)
#error You need a Windows NT 4.0 or later to run this stub because it uses these features:
#error -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 */
/* 8 */ NdrFcShort( 0x1c ),   /* x86 Stack size/offset = 28 */
/* 10 */ NdrFcShort( 0x0 ),   /* 0 */
/* 12 */ NdrFcShort( 0x8 ),   /* 8 */
/* 14 */ 0x7,                 /* Oi2 Flags: srv must size, clt must size, has return, */
                0x3,          /* 3 */

        /* Parameter txn_in */

/* 16 */ NdrFcShort( 0x8b ),   /* Flags: must size, must free, in, by val, */
/* 18 */ NdrFcShort( 0x4 ),   /* x86 Stack size/offset = 4 */
/* 20 */ NdrFcShort( 0x3e2 ), /* Type Offset=994 */

        /* Parameter txn_out */

/* 22 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=16 */
/* 24 */ NdrFcShort( 0x14 ),   /* x86 Stack size/offset = 20 */
/* 26 */ NdrFcShort( 0x3f4 ), /* Type Offset=1012 */

        /* Return value */

/* 28 */ NdrFcShort( 0x70 ),   /* Flags: out, return, base type, */
/* 30 */ NdrFcShort( 0x18 ),   /* x86 Stack size/offset = 24 */

```

```

/* 32 */ 0x8,          /* FC_LONG */
                0x0,          /* 0 */

/* Procedure Payment */

/* 34 */ 0x33,          /* FC_AUTO_HANDLE */
                0x6c,          /* Old Flags: object, Oi2 */
/* 36 */ NdrFcLong( 0x0 ),      /* 0 */
/* 40 */ NdrFcShort( 0x4 ),      /* 4 */
/* 42 */ NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28 */
/* 44 */ NdrFcShort( 0x0 ),      /* 0 */
/* 46 */ NdrFcShort( 0x8 ),      /* 8 */
/* 48 */ 0x7,          /* Oi2 Flags: srv must size, clt must size, has return, */
                0x3,          /* 3 */

/* Parameter txn_in */

/* 50 */ NdrFcShort( 0x8b ),     /* Flags: must size, must free, in, by val, */
/* 52 */ NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 54 */ NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

/* Parameter txn_out */

/* 56 */ NdrFcShort( 0x4113 ),   /* Flags: must size, must free, out, simple ref, srv alloc size=16 */
/* 58 */ NdrFcShort( 0x14 ),     /* x86 Stack size/offset = 20 */
/* 60 */ NdrFcShort( 0x3f4 ),    /* Type Offset=1012 */

/* Return value */

/* 62 */ NdrFcShort( 0x70 ),     /* Flags: out, return, base type, */
/* 64 */ NdrFcShort( 0x18 ),     /* x86 Stack size/offset = 24 */
/* 66 */ 0x8,          /* FC_LONG */
                0x0,          /* 0 */

/* Procedure Delivery */

/* 68 */ 0x33,          /* FC_AUTO_HANDLE */
                0x6c,          /* Old Flags: object, Oi2 */
/* 70 */ NdrFcLong( 0x0 ),      /* 0 */
/* 74 */ NdrFcShort( 0x5 ),      /* 5 */
/* 76 */ NdrFcShort( 0x1c ),     /* x86 Stack size/offset = 28 */
/* 78 */ NdrFcShort( 0x0 ),      /* 0 */
/* 80 */ NdrFcShort( 0x8 ),      /* 8 */
/* 82 */ 0x7,          /* Oi2 Flags: srv must size, clt must size, has return, */
                0x3,          /* 3 */

/* Parameter txn_in */

/* 84 */ NdrFcShort( 0x8b ),     /* Flags: must size, must free, in, by val, */
/* 86 */ NdrFcShort( 0x4 ),      /* x86 Stack size/offset = 4 */
/* 88 */ NdrFcShort( 0x3e2 ),    /* Type Offset=994 */

/* Parameter txn_out */

```



```

/* 90 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=16 */
/* 92 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
/* 94 */ NdrFcShort( 0x3f4 ), /* Type Offset=1012 */

```

```

/* Return value */

```

```

/* 96 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 98 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
/* 100 */ 0x8, /* FC_LONG */
0x0, /* 0 */

```

```

/* Procedure StockLevel */

```

```

/* 102 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 104 */ NdrFcLong( 0x0 ), /* 0 */
/* 108 */ NdrFcShort( 0x6 ), /* 6 */
/* 110 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */
/* 114 */ NdrFcShort( 0x8 ), /* 8 */
/* 116 */ 0x7, /* Oi2 Flags: srv must size, clt must size, has return, */
0x3, /* 3 */

```

```

/* Parameter txn_in */

```

```

/* 118 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 120 */ NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 122 */ NdrFcShort( 0x3e2 ), /* Type Offset=994 */

```

```

/* Parameter txn_out */

```

```

/* 124 */ NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple ref, srv alloc
size=16 */
/* 126 */ NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
/* 128 */ NdrFcShort( 0x3f4 ), /* Type Offset=1012 */

```

```

/* Return value */

```

```

/* 130 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 132 */ NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
/* 134 */ 0x8, /* FC_LONG */
0x0, /* 0 */

```

```

/* Procedure OrderStatus */

```

```

/* 136 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 138 */ NdrFcLong( 0x0 ), /* 0 */
/* 142 */ NdrFcShort( 0x7 ), /* 7 */
/* 144 */ NdrFcShort( 0x1c ), /* x86 Stack size/offset = 28 */
/* 146 */ NdrFcShort( 0x0 ), /* 0 */
/* 148 */ NdrFcShort( 0x8 ), /* 8 */

```

```

/* 150 */      0x7,          /* Oi2 Flags: srv must size, clt must size, has return, */
                0x3,          /* 3 */

        /* Parameter txn_in */

/* 152 */      NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 154 */      NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 156 */      NdrFcShort( 0x3e2 ), /* Type Offset=994 */

        /* Parameter txn_out */

/* 158 */      NdrFcShort( 0x4113 ), /* Flags: must size, must free, out, simple ref, srv alloc
size=16 */
/* 160 */      NdrFcShort( 0x14 ), /* x86 Stack size/offset = 20 */
/* 162 */      NdrFcShort( 0x3f4 ), /* Type Offset=1012 */

        /* Return value */

/* 164 */      NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 166 */      NdrFcShort( 0x18 ), /* x86 Stack size/offset = 24 */
/* 168 */      0x8,          /* FC_LONG */
                0x0,          /* 0 */

        /* Procedure CallSetComplete */

/* 170 */      0x33,          /* FC_AUTO_HANDLE */
                0x6c,          /* Old Flags: object, Oi2 */
/* 172 */      NdrFcLong( 0x0 ), /* 0 */
/* 176 */      NdrFcShort( 0x8 ), /* 8 */
/* 178 */      NdrFcShort( 0x8 ), /* x86 Stack size/offset = 8 */
/* 180 */      NdrFcShort( 0x0 ), /* 0 */
/* 182 */      NdrFcShort( 0x8 ), /* 8 */
/* 184 */      0x4,          /* Oi2 Flags: has return, */
                0x1,          /* 1 */

        /* Return value */

/* 186 */      NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 188 */      NdrFcShort( 0x4 ), /* x86 Stack size/offset = 4 */
/* 190 */      0x8,          /* FC_LONG */
                0x0,          /* 0 */

                0x0

    }
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /* 0 */
/* 2 */
        0x12, 0x0, /* FC_UP */

```

```

/* 4 */ NdrFcShort( 0x3ca ), /* Offset= 970 (974) */
/* 6 */
                                0x2b, /* FC_NON_ENCAPSULATED_UNION */
                                0x9, /* FC_ULONG */
/* 8 */ 0x7, /* Corr desc: FC_USHORT */
                                0x0, /* */
/* 10 */ NdrFcShort( 0xffff8 ), /* -8 */
/* 12 */ NdrFcShort( 0x2 ), /* Offset= 2 (14) */
/* 14 */ NdrFcShort( 0x10 ), /* 16 */
/* 16 */ NdrFcShort( 0x2f ), /* 47 */
/* 18 */ NdrFcLong( 0x14 ), /* 20 */
/* 22 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 24 */ NdrFcLong( 0x3 ), /* 3 */
/* 28 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 30 */ NdrFcLong( 0x11 ), /* 17 */
/* 34 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 36 */ NdrFcLong( 0x2 ), /* 2 */
/* 40 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 42 */ NdrFcLong( 0x4 ), /* 4 */
/* 46 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 48 */ NdrFcLong( 0x5 ), /* 5 */
/* 52 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 54 */ NdrFcLong( 0xb ), /* 11 */
/* 58 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 60 */ NdrFcLong( 0xa ), /* 10 */
/* 64 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 66 */ NdrFcLong( 0x6 ), /* 6 */
/* 70 */ NdrFcShort( 0xe8 ), /* Offset= 232 (302) */
/* 72 */ NdrFcLong( 0x7 ), /* 7 */
/* 76 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 78 */ NdrFcLong( 0x8 ), /* 8 */
/* 82 */ NdrFcShort( 0xe2 ), /* Offset= 226 (308) */
/* 84 */ NdrFcLong( 0xd ), /* 13 */
/* 88 */ NdrFcShort( 0xf4 ), /* Offset= 244 (332) */
/* 90 */ NdrFcLong( 0x9 ), /* 9 */
/* 94 */ NdrFcShort( 0x100 ), /* Offset= 256 (350) */
/* 96 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 100 */ NdrFcShort( 0x10c ), /* Offset= 268 (368) */
/* 102 */ NdrFcLong( 0x24 ), /* 36 */
/* 106 */ NdrFcShort( 0x31a ), /* Offset= 794 (900) */
/* 108 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 112 */ NdrFcShort( 0x314 ), /* Offset= 788 (900) */
/* 114 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 118 */ NdrFcShort( 0x312 ), /* Offset= 786 (904) */
/* 120 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 124 */ NdrFcShort( 0x310 ), /* Offset= 784 (908) */
/* 126 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 130 */ NdrFcShort( 0x30e ), /* Offset= 782 (912) */
/* 132 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 136 */ NdrFcShort( 0x30c ), /* Offset= 780 (916) */
/* 138 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 142 */ NdrFcShort( 0x30a ), /* Offset= 778 (920) */
/* 144 */ NdrFcLong( 0x4005 ), /* 16389 */

```

```

/* 148 */      NdrFcShort( 0x308 ), /* Offset= 776 (924) */
/* 150 */      NdrFcLong( 0x400b ), /* 16395 */
/* 154 */      NdrFcShort( 0x2f2 ), /* Offset= 754 (908) */
/* 156 */      NdrFcLong( 0x400a ), /* 16394 */
/* 160 */      NdrFcShort( 0x2f0 ), /* Offset= 752 (912) */
/* 162 */      NdrFcLong( 0x4006 ), /* 16390 */
/* 166 */      NdrFcShort( 0x2fa ), /* Offset= 762 (928) */
/* 168 */      NdrFcLong( 0x4007 ), /* 16391 */
/* 172 */      NdrFcShort( 0x2f0 ), /* Offset= 752 (924) */
/* 174 */      NdrFcLong( 0x4008 ), /* 16392 */
/* 178 */      NdrFcShort( 0x2f2 ), /* Offset= 754 (932) */
/* 180 */      NdrFcLong( 0x400d ), /* 16397 */
/* 184 */      NdrFcShort( 0x2f0 ), /* Offset= 752 (936) */
/* 186 */      NdrFcLong( 0x4009 ), /* 16393 */
/* 190 */      NdrFcShort( 0x2ee ), /* Offset= 750 (940) */
/* 192 */      NdrFcLong( 0x6000 ), /* 24576 */
/* 196 */      NdrFcShort( 0x2ec ), /* Offset= 748 (944) */
/* 198 */      NdrFcLong( 0x400c ), /* 16396 */
/* 202 */      NdrFcShort( 0x2ea ), /* Offset= 746 (948) */
/* 204 */      NdrFcLong( 0x10 ), /* 16 */
/* 208 */      NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 210 */      NdrFcLong( 0x12 ), /* 18 */
/* 214 */      NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 216 */      NdrFcLong( 0x13 ), /* 19 */
/* 220 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 222 */      NdrFcLong( 0x15 ), /* 21 */
/* 226 */      NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 228 */      NdrFcLong( 0x16 ), /* 22 */
/* 232 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 234 */      NdrFcLong( 0x17 ), /* 23 */
/* 238 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 240 */      NdrFcLong( 0xe ), /* 14 */
/* 244 */      NdrFcShort( 0x2c8 ), /* Offset= 712 (956) */
/* 246 */      NdrFcLong( 0x400e ), /* 16398 */
/* 250 */      NdrFcShort( 0x2cc ), /* Offset= 716 (966) */
/* 252 */      NdrFcLong( 0x4010 ), /* 16400 */
/* 256 */      NdrFcShort( 0x2ca ), /* Offset= 714 (970) */
/* 258 */      NdrFcLong( 0x4012 ), /* 16402 */
/* 262 */      NdrFcShort( 0x286 ), /* Offset= 646 (908) */
/* 264 */      NdrFcLong( 0x4013 ), /* 16403 */
/* 268 */      NdrFcShort( 0x284 ), /* Offset= 644 (912) */
/* 270 */      NdrFcLong( 0x4015 ), /* 16405 */
/* 274 */      NdrFcShort( 0x282 ), /* Offset= 642 (916) */
/* 276 */      NdrFcLong( 0x4016 ), /* 16406 */
/* 280 */      NdrFcShort( 0x278 ), /* Offset= 632 (912) */
/* 282 */      NdrFcLong( 0x4017 ), /* 16407 */
/* 286 */      NdrFcShort( 0x272 ), /* Offset= 626 (912) */
/* 288 */      NdrFcLong( 0x0 ), /* 0 */
/* 292 */      NdrFcShort( 0x0 ), /* Offset= 0 (292) */
/* 294 */      NdrFcLong( 0x1 ), /* 1 */
/* 298 */      NdrFcShort( 0x0 ), /* Offset= 0 (298) */
/* 300 */      NdrFcShort( 0xffffffff ), /* Offset= -1 (299) */
/* 302 */

```

```

0x15,          /* FC_STRUCT */
0x7,          /* 7 */
/* 304 */ NdrFcShort( 0x8 ), /* 8 */
/* 306 */ 0xb,          /* FC_HYPER */
0x5b,        /* FC_END */
/* 308 */
0x12, 0x0,    /* FC_UP */
/* 310 */ NdrFcShort( 0xc ), /* Offset= 12 (322) */
/* 312 */
0x1b,        /* FC_CARRAY */
0x1,        /* 1 */
/* 314 */ NdrFcShort( 0x2 ), /* 2 */
/* 316 */ 0x9,          /* Corr desc: FC_ULONG */
0x0,        /* */
/* 318 */ NdrFcShort( 0xffc ), /* -4 */
/* 320 */ 0x6,          /* FC_SHORT */
0x5b,        /* FC_END */
/* 322 */
0x17,        /* FC_CSTRUCT */
0x3,        /* 3 */
/* 324 */ NdrFcShort( 0x8 ), /* 8 */
/* 326 */ NdrFcShort( 0xfffff2 ), /* Offset= -14 (312) */
/* 328 */ 0x8,          /* FC_LONG */
0x8,        /* FC_LONG */
/* 330 */ 0x5c,        /* FC_PAD */
0x5b,        /* FC_END */
/* 332 */
0x2f,        /* FC_IP */
0x5a,        /* FC_CONSTANT_IID */
/* 334 */ NdrFcLong( 0x0 ), /* 0 */
/* 338 */ NdrFcShort( 0x0 ), /* 0 */
/* 340 */ NdrFcShort( 0x0 ), /* 0 */
/* 342 */ 0xc0,        /* 192 */
0x0,        /* 0 */
/* 344 */ 0x0,        /* 0 */
0x0,        /* 0 */
/* 346 */ 0x0,        /* 0 */
0x0,        /* 0 */
/* 348 */ 0x0,        /* 0 */
0x46,        /* 70 */
/* 350 */
0x2f,        /* FC_IP */
0x5a,        /* FC_CONSTANT_IID */
/* 352 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 356 */ NdrFcShort( 0x0 ), /* 0 */
/* 358 */ NdrFcShort( 0x0 ), /* 0 */
/* 360 */ 0xc0,        /* 192 */
0x0,        /* 0 */
/* 362 */ 0x0,        /* 0 */
0x0,        /* 0 */
/* 364 */ 0x0,        /* 0 */
0x0,        /* 0 */
/* 366 */ 0x0,        /* 0 */

```

```

0x46, /* 70 */
/* 368 */
0x12, 0x10, /* FC_UP [pointer_deref] */
/* 370 */ NdrFcShort( 0x2 ), /* Offset= 2 (372) */
/* 372 */
0x12, 0x0, /* FC_UP */
/* 374 */ NdrFcShort( 0x1fc ), /* Offset= 508 (882) */
/* 376 */
0x2a, /* FC_ENCAPSULATED_UNION */
0x49, /* 73 */
/* 378 */ NdrFcShort( 0x18 ), /* 24 */
/* 380 */ NdrFcShort( 0xa ), /* 10 */
/* 382 */ NdrFcLong( 0x8 ), /* 8 */
/* 386 */ NdrFcShort( 0x58 ), /* Offset= 88 (474) */
/* 388 */ NdrFcLong( 0xd ), /* 13 */
/* 392 */ NdrFcShort( 0x78 ), /* Offset= 120 (512) */
/* 394 */ NdrFcLong( 0x9 ), /* 9 */
/* 398 */ NdrFcShort( 0x94 ), /* Offset= 148 (546) */
/* 400 */ NdrFcLong( 0xc ), /* 12 */
/* 404 */ NdrFcShort( 0xbc ), /* Offset= 188 (592) */
/* 406 */ NdrFcLong( 0x24 ), /* 36 */
/* 410 */ NdrFcShort( 0x114 ), /* Offset= 276 (686) */
/* 412 */ NdrFcLong( 0x800d ), /* 32781 */
/* 416 */ NdrFcShort( 0x130 ), /* Offset= 304 (720) */
/* 418 */ NdrFcLong( 0x10 ), /* 16 */
/* 422 */ NdrFcShort( 0x148 ), /* Offset= 328 (750) */
/* 424 */ NdrFcLong( 0x2 ), /* 2 */
/* 428 */ NdrFcShort( 0x160 ), /* Offset= 352 (780) */
/* 430 */ NdrFcLong( 0x3 ), /* 3 */
/* 434 */ NdrFcShort( 0x178 ), /* Offset= 376 (810) */
/* 436 */ NdrFcLong( 0x14 ), /* 20 */
/* 440 */ NdrFcShort( 0x190 ), /* Offset= 400 (840) */
/* 442 */ NdrFcShort( 0xffffffff ), /* Offset= -1 (441) */
/* 444 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */
/* 446 */ NdrFcShort( 0x4 ), /* 4 */
/* 448 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 450 */ NdrFcShort( 0x0 ), /* 0 */
/* 452 */
0x4b, /* FC_PP */
0x5c, /* FC_PAD */
/* 454 */
0x48, /* FC_VARIABLE_REPEAT */
0x49, /* FC_FIXED_OFFSET */
/* 456 */ NdrFcShort( 0x4 ), /* 4 */
/* 458 */ NdrFcShort( 0x0 ), /* 0 */
/* 460 */ NdrFcShort( 0x1 ), /* 1 */
/* 462 */ NdrFcShort( 0x0 ), /* 0 */
/* 464 */ NdrFcShort( 0x0 ), /* 0 */
/* 466 */ 0x12, 0x0, /* FC_UP */
/* 468 */ NdrFcShort( 0xfffff6e ), /* Offset= -146 (322) */

```

```

/* 470 */
    0x5b,          /* FC_END */

    0x8,          /* FC_LONG */
/* 472 */ 0x5c,          /* FC_PAD */
    0x5b,          /* FC_END */
/* 474 */
    0x16,         /* FC_PSTRUCT */
    0x3,         /* 3 */
/* 476 */ NdrFcShort( 0x8 ), /* 8 */
/* 478 */
    0x4b,         /* FC_PP */
    0x5c,         /* FC_PAD */
/* 480 */
    0x46,         /* FC_NO_REPEAT */
    0x5c,         /* FC_PAD */
/* 482 */ NdrFcShort( 0x4 ), /* 4 */
/* 484 */ NdrFcShort( 0x4 ), /* 4 */
/* 486 */ 0x11, 0x0, /* FC_RP */
/* 488 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (444) */
/* 490 */
    0x5b,          /* FC_END */

    0x8,          /* FC_LONG */
/* 492 */ 0x8,          /* FC_LONG */
    0x5b,          /* FC_END */
/* 494 */
    0x21,         /* FC_BOGUS_ARRAY */
    0x3,         /* 3 */
/* 496 */ NdrFcShort( 0x0 ), /* 0 */
/* 498 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
    0x0,          /* */
/* 500 */ NdrFcShort( 0x0 ), /* 0 */
/* 502 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 506 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
    0x0,          /* 0 */
/* 508 */ NdrFcShort( 0xfffff50 ), /* Offset= -176 (332) */
/* 510 */ 0x5c,          /* FC_PAD */
    0x5b,          /* FC_END */
/* 512 */
    0x1a,         /* FC_BOGUS_STRUCT */
    0x3,         /* 3 */
/* 514 */ NdrFcShort( 0x8 ), /* 8 */
/* 516 */ NdrFcShort( 0x0 ), /* 0 */
/* 518 */ NdrFcShort( 0x6 ), /* Offset= 6 (524) */
/* 520 */ 0x8,          /* FC_LONG */
    0x36,         /* FC_POINTER */
/* 522 */ 0x5c,          /* FC_PAD */
    0x5b,          /* FC_END */
/* 524 */
    0x11, 0x0,    /* FC_RP */
/* 526 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (494) */
/* 528 */

```

```

        0x21,          /* FC_BOGUS_ARRAY */
        0x3,          /* 3 */
/* 530 */ NdrFcShort( 0x0 ), /* 0 */
/* 532 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
        0x0,          /* */
/* 534 */ NdrFcShort( 0x0 ), /* 0 */
/* 536 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 540 */ 0x4c,          /* FC_EMBEDDED_COMPLEX */
        0x0,          /* 0 */
/* 542 */ NdrFcShort( 0xfffff40 ), /* Offset= -192 (350) */
/* 544 */ 0x5c,          /* FC_PAD */
        0x5b,          /* FC_END */
/* 546 */
        0x1a,          /* FC_BOGUS_STRUCT */
        0x3,          /* 3 */
/* 548 */ NdrFcShort( 0x8 ), /* 8 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8,          /* FC_LONG */
        0x36,          /* FC_POINTER */
/* 556 */ 0x5c,          /* FC_PAD */
        0x5b,          /* FC_END */
/* 558 */
        0x11, 0x0,     /* FC_RP */
/* 560 */ NdrFcShort( 0xfffffe0 ), /* Offset= -32 (528) */
/* 562 */
        0x1b,          /* FC_CARRAY */
        0x3,          /* 3 */
/* 564 */ NdrFcShort( 0x4 ), /* 4 */
/* 566 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
        0x0,          /* */
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */
        0x4b,          /* FC_PP */
        0x5c,          /* FC_PAD */
/* 572 */
        0x48,          /* FC_VARIABLE_REPEAT */
        0x49,          /* FC_FIXED_OFFSET */
/* 574 */ NdrFcShort( 0x4 ), /* 4 */
/* 576 */ NdrFcShort( 0x0 ), /* 0 */
/* 578 */ NdrFcShort( 0x1 ), /* 1 */
/* 580 */ NdrFcShort( 0x0 ), /* 0 */
/* 582 */ NdrFcShort( 0x0 ), /* 0 */
/* 584 */ 0x12, 0x0,     /* FC_UP */
/* 586 */ NdrFcShort( 0x184 ), /* Offset= 388 (974) */
/* 588 */
        0x5b,          /* FC_END */
        0x8,          /* FC_LONG */
/* 590 */ 0x5c,          /* FC_PAD */
        0x5b,          /* FC_END */
/* 592 */
        0x1a,          /* FC_BOGUS_STRUCT */

```



```

0x3, /* 3 */
/* 594 */ NdrFcShort( 0x8 ), /* 8 */
/* 596 */ NdrFcShort( 0x0 ), /* 0 */
/* 598 */ NdrFcShort( 0x6 ), /* Offset= 6 (604) */
/* 600 */ 0x8, /* FC_LONG */
0x36, /* FC_POINTER */
/* 602 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 604 */
0x11, 0x0, /* FC_RP */
/* 606 */ NdrFcShort( 0xfffffd4 ), /* Offset= -44 (562) */
/* 608 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 610 */ NdrFcLong( 0x2f ), /* 47 */
/* 614 */ NdrFcShort( 0x0 ), /* 0 */
/* 616 */ NdrFcShort( 0x0 ), /* 0 */
/* 618 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 620 */ 0x0, /* 0 */
0x0, /* 0 */
/* 622 */ 0x0, /* 0 */
0x0, /* 0 */
/* 624 */ 0x0, /* 0 */
0x46, /* 70 */
/* 626 */
0x1b, /* FC_CARRAY */
0x0, /* 0 */
/* 628 */ NdrFcShort( 0x1 ), /* 1 */
/* 630 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 632 */ NdrFcShort( 0x4 ), /* 4 */
/* 634 */ 0x1, /* FC_BYTE */
0x5b, /* FC_END */
/* 636 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 638 */ NdrFcShort( 0x10 ), /* 16 */
/* 640 */ NdrFcShort( 0x0 ), /* 0 */
/* 642 */ NdrFcShort( 0xa ), /* Offset= 10 (652) */
/* 644 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 646 */ 0x4c, /* FC_EMBEDDED_COMPLEX */
0x0, /* 0 */
/* 648 */ NdrFcShort( 0xfffffd8 ), /* Offset= -40 (608) */
/* 650 */ 0x36, /* FC_POINTER */
0x5b, /* FC_END */
/* 652 */
0x12, 0x0, /* FC_UP */
/* 654 */ NdrFcShort( 0xfffffe4 ), /* Offset= -28 (626) */
/* 656 */
0x1b, /* FC_CARRAY */
0x3, /* 3 */

```

```

/* 658 */      NdrFcShort( 0x4 ),      /* 4 */
/* 660 */      0x19,      /* Corr desc: field pointer, FC_ULONG */
                0x0,      /* */
/* 662 */      NdrFcShort( 0x0 ),      /* 0 */
/* 664 */
                0x4b,      /* FC_PP */
                0x5c,      /* FC_PAD */
/* 666 */
                0x48,      /* FC_VARIABLE_REPEAT */
                0x49,      /* FC_FIXED_OFFSET */
/* 668 */      NdrFcShort( 0x4 ),      /* 4 */
/* 670 */      NdrFcShort( 0x0 ),      /* 0 */
/* 672 */      NdrFcShort( 0x1 ),      /* 1 */
/* 674 */      NdrFcShort( 0x0 ),      /* 0 */
/* 676 */      NdrFcShort( 0x0 ),      /* 0 */
/* 678 */      0x12, 0x0,      /* FC_UP */
/* 680 */      NdrFcShort( 0xfffffd4 ), /* Offset= -44 (636) */
/* 682 */
                0x5b,      /* FC_END */
                0x8,      /* FC_LONG */
/* 684 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 686 */
                0x1a,      /* FC_BOGUS_STRUCT */
                0x3,      /* 3 */
/* 688 */      NdrFcShort( 0x8 ),      /* 8 */
/* 690 */      NdrFcShort( 0x0 ),      /* 0 */
/* 692 */      NdrFcShort( 0x6 ),      /* Offset= 6 (698) */
/* 694 */      0x8,      /* FC_LONG */
                0x36,      /* FC_POINTER */
/* 696 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 698 */
                0x11, 0x0,      /* FC_RP */
/* 700 */      NdrFcShort( 0xfffffd4 ), /* Offset= -44 (656) */
/* 702 */
                0x1d,      /* FC_SMFARRAY */
                0x0,      /* 0 */
/* 704 */      NdrFcShort( 0x8 ),      /* 8 */
/* 706 */      0x1,      /* FC_BYTE */
                0x5b,      /* FC_END */
/* 708 */
                0x15,      /* FC_STRUCT */
                0x3,      /* 3 */
/* 710 */      NdrFcShort( 0x10 ),      /* 16 */
/* 712 */      0x8,      /* FC_LONG */
                0x6,      /* FC_SHORT */
/* 714 */      0x6,      /* FC_SHORT */
                0x4c,      /* FC_EMBEDDED_COMPLEX */
/* 716 */      0x0,      /* 0 */
                NdrFcShort( 0xfffff1 ), /* Offset= -15 (702) */
                0x5b,      /* FC_END */

```

```

/* 720 */
    0x1a,          /* FC_BOGUS_STRUCT */
    0x3,          /* 3 */
/* 722 */
NdrFcShort( 0x18 ), /* 24 */
/* 724 */
NdrFcShort( 0x0 ), /* 0 */
/* 726 */
NdrFcShort( 0xa ), /* Offset= 10 (736) */
/* 728 */
0x8,            /* FC_LONG */
    0x36,        /* FC_POINTER */
/* 730 */
0x4c,          /* FC_EMBEDDED_COMPLEX */
    0x0,          /* 0 */
/* 732 */
NdrFcShort( 0xfffffe8 ), /* Offset= -24 (708) */
/* 734 */
0x5c,          /* FC_PAD */
    0x5b,        /* FC_END */
/* 736 */
    0x11, 0x0,   /* FC_RP */
/* 738 */
NdrFcShort( 0xfffff0c ), /* Offset= -244 (494) */
/* 740 */
    0x1b,        /* FC_CARRAY */
    0x0,          /* 0 */
/* 742 */
NdrFcShort( 0x1 ), /* 1 */
/* 744 */
0x19,          /* Corr desc: field pointer, FC_ULONG */
    0x0,          /* */
/* 746 */
NdrFcShort( 0x0 ), /* 0 */
/* 748 */
0x1,           /* FC_BYTE */
    0x5b,        /* FC_END */
/* 750 */
    0x16,        /* FC_PSTRUCT */
    0x3,          /* 3 */
/* 752 */
NdrFcShort( 0x8 ), /* 8 */
/* 754 */
    0x4b,        /* FC_PP */
    0x5c,        /* FC_PAD */
/* 756 */
    0x46,        /* FC_NO_REPEAT */
    0x5c,        /* FC_PAD */
/* 758 */
NdrFcShort( 0x4 ), /* 4 */
/* 760 */
NdrFcShort( 0x4 ), /* 4 */
/* 762 */
0x12, 0x0,     /* FC_UP */
/* 764 */
NdrFcShort( 0xfffffe8 ), /* Offset= -24 (740) */
/* 766 */
    0x5b,        /* FC_END */
    0x8,          /* FC_LONG */
/* 768 */
0x8,           /* FC_LONG */
    0x5b,        /* FC_END */
/* 770 */
    0x1b,        /* FC_CARRAY */
    0x1,          /* 1 */
/* 772 */
NdrFcShort( 0x2 ), /* 2 */
/* 774 */
0x19,          /* Corr desc: field pointer, FC_ULONG */
    0x0,          /* */
/* 776 */
NdrFcShort( 0x0 ), /* 0 */
/* 778 */
0x6,           /* FC_SHORT */

```

```

                                0x5b,          /* FC_END */
/* 780 */
                                0x16,          /* FC_PSTRUCT */
                                0x3,          /* 3 */
/* 782 */ NdrFcShort( 0x8 ),          /* 8 */
/* 784 */
                                0x4b,          /* FC_PP */
                                0x5c,          /* FC_PAD */
/* 786 */
                                0x46,          /* FC_NO_REPEAT */
                                0x5c,          /* FC_PAD */
/* 788 */ NdrFcShort( 0x4 ),          /* 4 */
/* 790 */ NdrFcShort( 0x4 ),          /* 4 */
/* 792 */ 0x12, 0x0,          /* FC_UP */
/* 794 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24 (770) */
/* 796 */
                                0x5b,          /* FC_END */
/* 798 */
                                0x8,          /* FC_LONG */
0x8,          /* FC_LONG */
/* 800 */
                                0x5b,          /* FC_END */
/* 802 */
                                0x1b,          /* FC_CARRAY */
                                0x3,          /* 3 */
/* 804 */ NdrFcShort( 0x4 ),          /* 4 */
/* 804 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 806 */ NdrFcShort( 0x0 ),          /* 0 */
/* 808 */ 0x8,          /* FC_LONG */
/* 810 */
                                0x5b,          /* FC_END */
/* 812 */
                                0x16,          /* FC_PSTRUCT */
                                0x3,          /* 3 */
/* 814 */ NdrFcShort( 0x8 ),          /* 8 */
/* 816 */
                                0x4b,          /* FC_PP */
                                0x5c,          /* FC_PAD */
/* 818 */
                                0x46,          /* FC_NO_REPEAT */
                                0x5c,          /* FC_PAD */
/* 818 */ NdrFcShort( 0x4 ),          /* 4 */
/* 820 */ NdrFcShort( 0x4 ),          /* 4 */
/* 822 */ 0x12, 0x0,          /* FC_UP */
/* 824 */ NdrFcShort( 0xfffffe8 ), /* Offset= -24 (800) */
/* 826 */
                                0x5b,          /* FC_END */
/* 828 */
                                0x8,          /* FC_LONG */
0x8,          /* FC_LONG */
/* 830 */
                                0x5b,          /* FC_END */
/* 830 */
                                0x1b,          /* FC_CARRAY */
                                0x7,          /* 7 */

```

```

/* 832 */      NdrFcShort( 0x8 ),      /* 8 */
/* 834 */      0x19,      /* Corr desc: field pointer, FC_ULONG */
                0x0,      /* */
/* 836 */      NdrFcShort( 0x0 ),      /* 0 */
/* 838 */      0xb,      /* FC_HYPER */
                0x5b,      /* FC_END */
/* 840 */
                0x16,      /* FC_PSTRUCT */
                0x3,      /* 3 */
/* 842 */      NdrFcShort( 0x8 ),      /* 8 */
/* 844 */
                0x4b,      /* FC_PP */
                0x5c,      /* FC_PAD */
/* 846 */
                0x46,      /* FC_NO_REPEAT */
                0x5c,      /* FC_PAD */
/* 848 */      NdrFcShort( 0x4 ),      /* 4 */
/* 850 */      NdrFcShort( 0x4 ),      /* 4 */
/* 852 */      0x12, 0x0,      /* FC_UP */
/* 854 */      NdrFcShort( 0xfffffe8 ), /* Offset= -24 (830) */
/* 856 */
                0x5b,      /* FC_END */
/* 858 */
                0x8,      /* FC_LONG */
                0x8,      /* FC_LONG */
                0x5b,      /* FC_END */
/* 860 */
                0x15,      /* FC_STRUCT */
                0x3,      /* 3 */
/* 862 */      NdrFcShort( 0x8 ),      /* 8 */
/* 864 */      0x8,      /* FC_LONG */
                0x8,      /* FC_LONG */
/* 866 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 868 */
                0x1b,      /* FC_CARRAY */
                0x3,      /* 3 */
/* 870 */      NdrFcShort( 0x8 ),      /* 8 */
/* 872 */      0x7,      /* Corr desc: FC_USHORT */
                0x0,      /* */
/* 874 */      NdrFcShort( 0xffd8 ),      /* -40 */
/* 876 */      0x4c,      /* FC_EMBEDDED_COMPLEX */
                0x0,      /* 0 */
/* 878 */      NdrFcShort( 0xfffffee ), /* Offset= -18 (860) */
/* 880 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 882 */
                0x1a,      /* FC_BOGUS_STRUCT */
                0x3,      /* 3 */
/* 884 */      NdrFcShort( 0x28 ),      /* 40 */
/* 886 */      NdrFcShort( 0xfffffee ), /* Offset= -18 (868) */
/* 888 */      NdrFcShort( 0x0 ),      /* Offset= 0 (888) */
/* 890 */      0x6,      /* FC_SHORT */

```

```

/* 892 */      0x6,          /* FC_SHORT */
0x8,          /* FC_LONG */
/* 894 */      0x8,          /* FC_LONG */
0x4c,        /* FC_EMBEDDED_COMPLEX */
0x0,        /* 0 */
/* 896 */      NdrFcShort( 0xffffdf8 ), /* Offset= -520 (376) */
/* 898 */      0x5c,        /* FC_PAD */
0x5b,        /* FC_END */
/* 900 */
0x12, 0x0,   /* FC_UP */
/* 902 */      NdrFcShort( 0xffffef6 ), /* Offset= -266 (636) */
/* 904 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 906 */      0x1,          /* FC_BYTE */
0x5c,        /* FC_PAD */
/* 908 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 910 */      0x6,          /* FC_SHORT */
0x5c,        /* FC_PAD */
/* 912 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 914 */      0x8,          /* FC_LONG */
0x5c,        /* FC_PAD */
/* 916 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 918 */      0xb,          /* FC_HYPER */
0x5c,        /* FC_PAD */
/* 920 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 922 */      0xa,          /* FC_FLOAT */
0x5c,        /* FC_PAD */
/* 924 */
0x12, 0x8,   /* FC_UP [simple_pointer] */
/* 926 */      0xc,          /* FC_DOUBLE */
0x5c,        /* FC_PAD */
/* 928 */
0x12, 0x0,   /* FC_UP */
/* 930 */      NdrFcShort( 0xffffd8c ), /* Offset= -628 (302) */
/* 932 */
0x12, 0x10,  /* FC_UP [pointer_deref] */
/* 934 */      NdrFcShort( 0xffffd8e ), /* Offset= -626 (308) */
/* 936 */
0x12, 0x10,  /* FC_UP [pointer_deref] */
/* 938 */      NdrFcShort( 0xffffda2 ), /* Offset= -606 (332) */
/* 940 */
0x12, 0x10,  /* FC_UP [pointer_deref] */
/* 942 */      NdrFcShort( 0xffffdb0 ), /* Offset= -592 (350) */
/* 944 */
0x12, 0x10,  /* FC_UP [pointer_deref] */
/* 946 */      NdrFcShort( 0xffffdbe ), /* Offset= -578 (368) */
/* 948 */
0x12, 0x10,  /* FC_UP [pointer_deref] */
/* 950 */      NdrFcShort( 0x2 ), /* Offset= 2 (952) */

```

```

/* 952 */
        0x12, 0x0,      /* FC_UP */
/* 954 */ NdrFcShort( 0x14 ), /* Offset= 20 (974) */
/* 956 */
        0x15,      /* FC_STRUCT */
        0x7,      /* 7 */
/* 958 */ NdrFcShort( 0x10 ), /* 16 */
/* 960 */ 0x6,      /* FC_SHORT */
        0x1,      /* FC_BYTE */
/* 962 */ 0x1,      /* FC_BYTE */
        0x8,      /* FC_LONG */
/* 964 */ 0xb,      /* FC_HYPER */
        0x5b,     /* FC_END */
/* 966 */
        0x12, 0x0,      /* FC_UP */
/* 968 */ NdrFcShort( 0xffffffff ), /* Offset= -12 (956) */
/* 970 */
        0x12, 0x8,     /* FC_UP [simple_pointer] */
/* 972 */ 0x2,      /* FC_CHAR */
        0x5c,     /* FC_PAD */
/* 974 */
        0x1a,     /* FC_BOGUS_STRUCT */
        0x7,      /* 7 */
/* 976 */ NdrFcShort( 0x20 ), /* 32 */
/* 978 */ NdrFcShort( 0x0 ), /* 0 */
/* 980 */ NdrFcShort( 0x0 ), /* Offset= 0 (980) */
/* 982 */ 0x8,      /* FC_LONG */
        0x8,      /* FC_LONG */
/* 984 */ 0x6,      /* FC_SHORT */
        0x6,      /* FC_SHORT */
/* 986 */ 0x6,      /* FC_SHORT */
        0x6,      /* FC_SHORT */
/* 988 */ 0x4c,     /* FC_EMBEDDED_COMPLEX */
        0x0,      /* 0 */
/* 990 */ NdrFcShort( 0xffffc28 ), /* Offset= -984 (6) */
/* 992 */ 0x5c,     /* FC_PAD */
        0x5b,     /* FC_END */
/* 994 */ 0xb4,     /* FC_USER_MARSHAL */
        0x83,     /* 131 */
/* 996 */ NdrFcShort( 0x0 ), /* 0 */
/* 998 */ NdrFcShort( 0x10 ), /* 16 */
/* 1000 */ NdrFcShort( 0x0 ), /* 0 */
/* 1002 */ NdrFcShort( 0xffffc18 ), /* Offset= -1000 (2) */
/* 1004 */
        0x11, 0x4,     /* FC_RP [allocated_on_stack] */
/* 1006 */ NdrFcShort( 0x6 ), /* Offset= 6 (1012) */
/* 1008 */
        0x13, 0x0,     /* FC_OP */
/* 1010 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (974) */
/* 1012 */ 0xb4,     /* FC_USER_MARSHAL */
        0x83,     /* 131 */
/* 1014 */ NdrFcShort( 0x0 ), /* 0 */
/* 1016 */ NdrFcShort( 0x10 ), /* 16 */

```

```

/* 1018 */      NdrFcShort( 0x0 ),      /* 0 */
/* 1020 */      NdrFcShort( 0xffffffff4 ), /* Offset= -12 (1008) */

                0x0
            }
        };

static          const          USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
    {
        {
            VARIANT_UserSize
            ,VARIANT_UserMarshal
            ,VARIANT_UserUnmarshal
            ,VARIANT_UserFree
        }
    };

/* Standard interface: __MIDL_itf_tpsc_com_ps_0000, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
    {
        0,
        34,
        68,
        102,
        136,
        170
    };

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
    {
        &Object_StubDesc,
        __MIDL_ProcFormatString.Format,
        &ITPCC_FormatStringOffsetTable[-3],
        0,
        0,
        0
    };

```



```

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,
    0,
    _MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0,
    0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */ ,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */ ,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */ ,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */ ,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    _MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x20000, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
};

```

```

    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n) IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName *) &_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
}

```

```

};

#endif /* !defined(_M_IA64) && !defined(_M_AMD64)*/

#pragma warning( disable: 4049 ) /* more than 64k source lines */

/* this ALWAYS GENERATED file contains the proxy stub code */

/* File created by MIDL compiler version 6.00.0347 */
/* at Fri Apr 15 14:48:43 2005
*/
/* Compiler settings for .\src\tpcc_com_ps.idl:
   Oicf, W1, Zp8, env=Win64 (32b run,appending)
   protocol : dce , ms_ext, c_ext, robust
   error checks: allocation ref bounds_check enum stub_data
   VC __declspec() decoration level:
       __declspec(uuid()), __declspec(selectany), __declspec(novtable)
       DECLSPEC_UUID(), MIDL_INTERFACE()
*/
//@@MIDL_FILE_HEADING( )

#if defined(_M_IA64) || defined(_M_AMD64)
#define USE_STUBLESS_PROXY

/* verify that the <rpcproxy.h> version is high enough to compile this file*/
#ifndef __REDQ_RPCPROXY_H_VERSION__
#define __REQUIRED_RPCPROXY_H_VERSION__ 475
#endif

#include "rpcproxy.h"
#ifndef __RPCPROXY_H_VERSION__
#error this stub requires an updated version of <rpcproxy.h>
#endif // __RPCPROXY_H_VERSION__

#include "tpcc_com_ps.h"

#define TYPE_FORMAT_STRING_SIZE 1003
#define PROC_FORMAT_STRING_SIZE 253
#define TRANSMIT_AS_TABLE_SIZE 0
#define WIRE_MARSHAL_TABLE_SIZE 1

typedef struct _MIDL_TYPE_FORMAT_STRING
{
    short    Pad;
    unsigned char    Format[ TYPE_FORMAT_STRING_SIZE ];
} MIDL_TYPE_FORMAT_STRING;

```

```

typedef struct _MIDL_PROC_FORMAT_STRING
{
    short    Pad;
    unsigned char Format[ PROC_FORMAT_STRING_SIZE ];
} MIDL_PROC_FORMAT_STRING;

static RPC_SYNTAX_IDENTIFIER _RpcTransferSyntax =
{{0x8A885D04,0x1CEB,0x11C9,{0x9F,0xE8,0x08,0x00,0x2B,0x10,0x48,0x60}},{2,0}};

extern const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString;
extern const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString;

extern const MIDL_STUB_DESC Object_StubDesc;

extern const MIDL_SERVER_INFO ITPCC_ServerInfo;
extern const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo;

extern          const          USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ];

#if !defined(__RPC_WIN64__)
#error Invalid build platform for this stub.
#endif

static const MIDL_PROC_FORMAT_STRING __MIDL_ProcFormatString =
{
    0,
    {

        /* Procedure NewOrder */

                0x33,          /* FC_AUTO_HANDLE */
                0x6c,          /* Old Flags: object, Oi2 */
/* 2 */ NdrFcLong( 0x0 ),      /* 0 */
/* 6 */ NdrFcShort( 0x3 ),     /* 3 */
/* 8 */ NdrFcShort( 0x30 ),    /* ia64 Stack size/offset = 48 */
/* 10 */ NdrFcShort( 0x0 ),    /* 0 */
/* 12 */ NdrFcShort( 0x8 ),    /* 8 */
/* 14 */ 0x47,                 /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
                0x3,           /* 3 */
/* 16 */ 0xa,                  /* 10 */
                0x7,           /* Ext Flags: new corr desc, clt corr check, srv corr check, */
/*
/* 18 */ NdrFcShort( 0x20 ),    /* 32 */
/* 20 */ NdrFcShort( 0x20 ),    /* 32 */
/* 22 */ NdrFcShort( 0x0 ),     /* 0 */
/* 24 */ NdrFcShort( 0x0 ),     /* 0 */

```

```

    /* Parameter txn_in */

/* 26 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 28 */ NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 30 */ NdrFcShort( 0x3ce ), /* Type Offset=974 */

    /* Parameter txn_out */

/* 32 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=24 */
/* 34 */ NdrFcShort( 0x20 ), /* ia64 Stack size/offset = 32 */
/* 36 */ NdrFcShort( 0x3e0 ), /* Type Offset=992 */

    /* Return value */

/* 38 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 40 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
/* 42 */ 0x8, /* FC_LONG */
        0x0, /* 0 */

    /* Procedure Payment */

/* 44 */ 0x33, /* FC_AUTO_HANDLE */
        0x6c, /* Old Flags: object, Oi2 */
/* 46 */ NdrFcLong( 0x0 ), /* 0 */
/* 50 */ NdrFcShort( 0x4 ), /* 4 */
/* 52 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
/* 54 */ NdrFcShort( 0x0 ), /* 0 */
/* 56 */ NdrFcShort( 0x8 ), /* 8 */
/* 58 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
        0x3, /* 3 */
/* 60 */ 0xa, /* 10 */
        0x7, /* Ext Flags: new corr desc, clt corr check, srv corr check,
*/
/* 62 */ NdrFcShort( 0x20 ), /* 32 */
/* 64 */ NdrFcShort( 0x20 ), /* 32 */
/* 66 */ NdrFcShort( 0x0 ), /* 0 */
/* 68 */ NdrFcShort( 0x0 ), /* 0 */

    /* Parameter txn_in */

/* 70 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 72 */ NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 74 */ NdrFcShort( 0x3ce ), /* Type Offset=974 */

    /* Parameter txn_out */

/* 76 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple ref, srv alloc size=24 */
/* 78 */ NdrFcShort( 0x20 ), /* ia64 Stack size/offset = 32 */
/* 80 */ NdrFcShort( 0x3e0 ), /* Type Offset=992 */

    /* Return value */

/* 82 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */

```

```

/* 84 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
/* 86 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure Delivery */

/* 88 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 90 */ NdrFcLong( 0x0 ), /* 0 */
/* 94 */ NdrFcShort( 0x5 ), /* 5 */
/* 96 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
/* 98 */ NdrFcShort( 0x0 ), /* 0 */
/* 100 */ NdrFcShort( 0x8 ), /* 8 */
/* 102 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
0x3, /* 3 */
/* 104 */ 0xa, /* 10 */
0x7, /* Ext Flags: new corr desc, clt corr check, srv corr check,
*/
/* 106 */ NdrFcShort( 0x20 ), /* 32 */
/* 108 */ NdrFcShort( 0x20 ), /* 32 */
/* 110 */ NdrFcShort( 0x0 ), /* 0 */
/* 112 */ NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 114 */ NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 116 */ NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 118 */ NdrFcShort( 0x3ce ), /* Type Offset=974 */

/* Parameter txn_out */

/* 120 */ NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple ref, srv alloc
size=24 */
/* 122 */ NdrFcShort( 0x20 ), /* ia64 Stack size/offset = 32 */
/* 124 */ NdrFcShort( 0x3e0 ), /* Type Offset=992 */

/* Return value */

/* 126 */ NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 128 */ NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
/* 130 */ 0x8, /* FC_LONG */
0x0, /* 0 */

/* Procedure StockLevel */

/* 132 */ 0x33, /* FC_AUTO_HANDLE */
0x6c, /* Old Flags: object, Oi2 */
/* 134 */ NdrFcLong( 0x0 ), /* 0 */
/* 138 */ NdrFcShort( 0x6 ), /* 6 */
/* 140 */ NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
/* 142 */ NdrFcShort( 0x0 ), /* 0 */
/* 144 */ NdrFcShort( 0x8 ), /* 8 */
/* 146 */ 0x47, /* Oi2 Flags: srv must size, clt must size, has return, has ext, */

```

```

        0x3,          /* 3 */
/* 148 */      0xa,          /* 10 */
        0x7,          /* Ext Flags: new corr desc, clt corr check, srv corr check,
*/
/* 150 */      NdrFcShort( 0x20 ), /* 32 */
/* 152 */      NdrFcShort( 0x20 ), /* 32 */
/* 154 */      NdrFcShort( 0x0 ), /* 0 */
/* 156 */      NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 158 */      NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 160 */      NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 162 */      NdrFcShort( 0x3ce ), /* Type Offset=974 */

/* Parameter txn_out */

/* 164 */      NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple ref, srv alloc
size=24 */
/* 166 */      NdrFcShort( 0x20 ), /* ia64 Stack size/offset = 32 */
/* 168 */      NdrFcShort( 0x3e0 ), /* Type Offset=992 */

/* Return value */

/* 170 */      NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 172 */      NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
/* 174 */      0x8,          /* FC_LONG */
        0x0,          /* 0 */

/* Procedure OrderStatus */

/* 176 */      0x33,          /* FC_AUTO_HANDLE */
        0x6c,          /* Old Flags: object, Oi2 */
/* 178 */      NdrFcLong( 0x0 ), /* 0 */
/* 182 */      NdrFcShort( 0x7 ), /* 7 */
/* 184 */      NdrFcShort( 0x30 ), /* ia64 Stack size/offset = 48 */
/* 186 */      NdrFcShort( 0x0 ), /* 0 */
/* 188 */      NdrFcShort( 0x8 ), /* 8 */
/* 190 */      0x47,          /* Oi2 Flags: srv must size, clt must size, has return, has ext, */
        0x3,          /* 3 */
/* 192 */      0xa,          /* 10 */
        0x7,          /* Ext Flags: new corr desc, clt corr check, srv corr check,
*/
/* 194 */      NdrFcShort( 0x20 ), /* 32 */
/* 196 */      NdrFcShort( 0x20 ), /* 32 */
/* 198 */      NdrFcShort( 0x0 ), /* 0 */
/* 200 */      NdrFcShort( 0x0 ), /* 0 */

/* Parameter txn_in */

/* 202 */      NdrFcShort( 0x8b ), /* Flags: must size, must free, in, by val, */
/* 204 */      NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 206 */      NdrFcShort( 0x3ce ), /* Type Offset=974 */

```

```

        /* Parameter txn_out */

/* 208 */      NdrFcShort( 0x6113 ), /* Flags: must size, must free, out, simple ref, srv alloc
size=24 */
/* 210 */      NdrFcShort( 0x20 ), /* ia64 Stack size/offset = 32 */
/* 212 */      NdrFcShort( 0x3e0 ), /* Type Offset=992 */

        /* Return value */

/* 214 */      NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 216 */      NdrFcShort( 0x28 ), /* ia64 Stack size/offset = 40 */
/* 218 */      0x8, /* FC_LONG */
                0x0, /* 0 */

        /* Procedure CallSetComplete */

/* 220 */      0x33, /* FC_AUTO_HANDLE */
                0x6c, /* Old Flags: object, Oi2 */
/* 222 */      NdrFcLong( 0x0 ), /* 0 */
/* 226 */      NdrFcShort( 0x8 ), /* 8 */
/* 228 */      NdrFcShort( 0x10 ), /* ia64 Stack size/offset = 16 */
/* 230 */      NdrFcShort( 0x0 ), /* 0 */
/* 232 */      NdrFcShort( 0x8 ), /* 8 */
/* 234 */      0x44, /* Oi2 Flags: has return, has ext, */
                0x1, /* 1 */
/* 236 */      0xa, /* 10 */
                0x1, /* Ext Flags: new corr desc, */
/* 238 */      NdrFcShort( 0x0 ), /* 0 */
/* 240 */      NdrFcShort( 0x0 ), /* 0 */
/* 242 */      NdrFcShort( 0x0 ), /* 0 */
/* 244 */      NdrFcShort( 0x0 ), /* 0 */

        /* Return value */

/* 246 */      NdrFcShort( 0x70 ), /* Flags: out, return, base type, */
/* 248 */      NdrFcShort( 0x8 ), /* ia64 Stack size/offset = 8 */
/* 250 */      0x8, /* FC_LONG */
                0x0, /* 0 */

                0x0
    }
};

static const MIDL_TYPE_FORMAT_STRING __MIDL_TypeFormatString =
{
    0,
    {
        NdrFcShort( 0x0 ), /* 0 */
/* 2 */
        0x12, 0x0, /* FC_UP */
/* 4 */ NdrFcShort( 0x3b6 ), /* Offset= 950 (954) */
/* 6 */

```



```

0x2b,          /* FC_NON_ENCAPSULATED_UNION */
0x9,          /* FC_ULONG */
/* 8 */ 0x7,   /* Corr desc: FC_USHORT */
0x0,        /* */
/* 10 */ NdrFcShort( 0xffff ), /* -8 */
/* 12 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 14 */ NdrFcShort( 0x2 ), /* Offset= 2 (16) */
/* 16 */ NdrFcShort( 0x10 ), /* 16 */
/* 18 */ NdrFcShort( 0x2f ), /* 47 */
/* 20 */ NdrFcLong( 0x14 ), /* 20 */
/* 24 */ NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 26 */ NdrFcLong( 0x3 ), /* 3 */
/* 30 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 32 */ NdrFcLong( 0x11 ), /* 17 */
/* 36 */ NdrFcShort( 0x8001 ), /* Simple arm type: FC_BYTE */
/* 38 */ NdrFcLong( 0x2 ), /* 2 */
/* 42 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 44 */ NdrFcLong( 0x4 ), /* 4 */
/* 48 */ NdrFcShort( 0x800a ), /* Simple arm type: FC_FLOAT */
/* 50 */ NdrFcLong( 0x5 ), /* 5 */
/* 54 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 56 */ NdrFcLong( 0xb ), /* 11 */
/* 60 */ NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 62 */ NdrFcLong( 0xa ), /* 10 */
/* 66 */ NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 68 */ NdrFcLong( 0x6 ), /* 6 */
/* 72 */ NdrFcShort( 0xe8 ), /* Offset= 232 (304) */
/* 74 */ NdrFcLong( 0x7 ), /* 7 */
/* 78 */ NdrFcShort( 0x800c ), /* Simple arm type: FC_DOUBLE */
/* 80 */ NdrFcLong( 0x8 ), /* 8 */
/* 84 */ NdrFcShort( 0xe2 ), /* Offset= 226 (310) */
/* 86 */ NdrFcLong( 0xd ), /* 13 */
/* 90 */ NdrFcShort( 0xf6 ), /* Offset= 246 (336) */
/* 92 */ NdrFcLong( 0x9 ), /* 9 */
/* 96 */ NdrFcShort( 0x102 ), /* Offset= 258 (354) */
/* 98 */ NdrFcLong( 0x2000 ), /* 8192 */
/* 102 */ NdrFcShort( 0x10e ), /* Offset= 270 (372) */
/* 104 */ NdrFcLong( 0x24 ), /* 36 */
/* 108 */ NdrFcShort( 0x304 ), /* Offset= 772 (880) */
/* 110 */ NdrFcLong( 0x4024 ), /* 16420 */
/* 114 */ NdrFcShort( 0x2fe ), /* Offset= 766 (880) */
/* 116 */ NdrFcLong( 0x4011 ), /* 16401 */
/* 120 */ NdrFcShort( 0x2fc ), /* Offset= 764 (884) */
/* 122 */ NdrFcLong( 0x4002 ), /* 16386 */
/* 126 */ NdrFcShort( 0x2fa ), /* Offset= 762 (888) */
/* 128 */ NdrFcLong( 0x4003 ), /* 16387 */
/* 132 */ NdrFcShort( 0x2f8 ), /* Offset= 760 (892) */
/* 134 */ NdrFcLong( 0x4014 ), /* 16404 */
/* 138 */ NdrFcShort( 0x2f6 ), /* Offset= 758 (896) */
/* 140 */ NdrFcLong( 0x4004 ), /* 16388 */
/* 144 */ NdrFcShort( 0x2f4 ), /* Offset= 756 (900) */
/* 146 */ NdrFcLong( 0x4005 ), /* 16389 */
/* 150 */ NdrFcShort( 0x2f2 ), /* Offset= 754 (904) */

```

```

/* 152 */      NdrFcLong( 0x400b ), /* 16395 */
/* 156 */      NdrFcShort( 0x2dc ), /* Offset= 732 (888) */
/* 158 */      NdrFcLong( 0x400a ), /* 16394 */
/* 162 */      NdrFcShort( 0x2da ), /* Offset= 730 (892) */
/* 164 */      NdrFcLong( 0x4006 ), /* 16390 */
/* 168 */      NdrFcShort( 0x2e4 ), /* Offset= 740 (908) */
/* 170 */      NdrFcLong( 0x4007 ), /* 16391 */
/* 174 */      NdrFcShort( 0x2da ), /* Offset= 730 (904) */
/* 176 */      NdrFcLong( 0x4008 ), /* 16392 */
/* 180 */      NdrFcShort( 0x2dc ), /* Offset= 732 (912) */
/* 182 */      NdrFcLong( 0x400d ), /* 16397 */
/* 186 */      NdrFcShort( 0x2da ), /* Offset= 730 (916) */
/* 188 */      NdrFcLong( 0x4009 ), /* 16393 */
/* 192 */      NdrFcShort( 0x2d8 ), /* Offset= 728 (920) */
/* 194 */      NdrFcLong( 0x6000 ), /* 24576 */
/* 198 */      NdrFcShort( 0x2d6 ), /* Offset= 726 (924) */
/* 200 */      NdrFcLong( 0x400c ), /* 16396 */
/* 204 */      NdrFcShort( 0x2d4 ), /* Offset= 724 (928) */
/* 206 */      NdrFcLong( 0x10 ), /* 16 */
/* 210 */      NdrFcShort( 0x8002 ), /* Simple arm type: FC_CHAR */
/* 212 */      NdrFcLong( 0x12 ), /* 18 */
/* 216 */      NdrFcShort( 0x8006 ), /* Simple arm type: FC_SHORT */
/* 218 */      NdrFcLong( 0x13 ), /* 19 */
/* 222 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 224 */      NdrFcLong( 0x15 ), /* 21 */
/* 228 */      NdrFcShort( 0x800b ), /* Simple arm type: FC_HYPER */
/* 230 */      NdrFcLong( 0x16 ), /* 22 */
/* 234 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 236 */      NdrFcLong( 0x17 ), /* 23 */
/* 240 */      NdrFcShort( 0x8008 ), /* Simple arm type: FC_LONG */
/* 242 */      NdrFcLong( 0xe ), /* 14 */
/* 246 */      NdrFcShort( 0x2b2 ), /* Offset= 690 (936) */
/* 248 */      NdrFcLong( 0x400e ), /* 16398 */
/* 252 */      NdrFcShort( 0x2b6 ), /* Offset= 694 (946) */
/* 254 */      NdrFcLong( 0x4010 ), /* 16400 */
/* 258 */      NdrFcShort( 0x2b4 ), /* Offset= 692 (950) */
/* 260 */      NdrFcLong( 0x4012 ), /* 16402 */
/* 264 */      NdrFcShort( 0x270 ), /* Offset= 624 (888) */
/* 266 */      NdrFcLong( 0x4013 ), /* 16403 */
/* 270 */      NdrFcShort( 0x26e ), /* Offset= 622 (892) */
/* 272 */      NdrFcLong( 0x4015 ), /* 16405 */
/* 276 */      NdrFcShort( 0x26c ), /* Offset= 620 (896) */
/* 278 */      NdrFcLong( 0x4016 ), /* 16406 */
/* 282 */      NdrFcShort( 0x262 ), /* Offset= 610 (892) */
/* 284 */      NdrFcLong( 0x4017 ), /* 16407 */
/* 288 */      NdrFcShort( 0x25c ), /* Offset= 604 (892) */
/* 290 */      NdrFcLong( 0x0 ), /* 0 */
/* 294 */      NdrFcShort( 0x0 ), /* Offset= 0 (294) */
/* 296 */      NdrFcLong( 0x1 ), /* 1 */
/* 300 */      NdrFcShort( 0x0 ), /* Offset= 0 (300) */
/* 302 */      NdrFcShort( 0xffffffff ), /* Offset= -1 (301) */
/* 304 */

```

```

0x15, /* FC_STRUCT */

```

```

0x7, /* 7 */
/* 306 */ NdrFcShort( 0x8 ), /* 8 */
/* 308 */ 0xb, /* FC_HYPER */
0x5b, /* FC_END */
/* 310 */
0x12, 0x0, /* FC_UP */
/* 312 */ NdrFcShort( 0xe ), /* Offset= 14 (326) */
/* 314 */
0x1b, /* FC_CARRAY */
0x1, /* 1 */
/* 316 */ NdrFcShort( 0x2 ), /* 2 */
/* 318 */ 0x9, /* Corr desc: FC_ULONG */
0x0, /* */
/* 320 */ NdrFcShort( 0xfffc ), /* -4 */
/* 322 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 324 */ 0x6, /* FC_SHORT */
0x5b, /* FC_END */
/* 326 */
0x17, /* FC_CSTRUCT */
0x3, /* 3 */
/* 328 */ NdrFcShort( 0x8 ), /* 8 */
/* 330 */ NdrFcShort( 0xffffffff0 ), /* Offset= -16 (314) */
/* 332 */ 0x8, /* FC_LONG */
0x8, /* FC_LONG */
/* 334 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 336 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 338 */ NdrFcLong( 0x0 ), /* 0 */
/* 342 */ NdrFcShort( 0x0 ), /* 0 */
/* 344 */ NdrFcShort( 0x0 ), /* 0 */
/* 346 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 348 */ 0x0, /* 0 */
0x0, /* 0 */
/* 350 */ 0x0, /* 0 */
0x0, /* 0 */
/* 352 */ 0x0, /* 0 */
0x46, /* 70 */
/* 354 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 356 */ NdrFcLong( 0x20400 ), /* 132096 */
/* 360 */ NdrFcShort( 0x0 ), /* 0 */
/* 362 */ NdrFcShort( 0x0 ), /* 0 */
/* 364 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 366 */ 0x0, /* 0 */
0x0, /* 0 */
/* 368 */ 0x0, /* 0 */
0x0, /* 0 */
/* 370 */ 0x0, /* 0 */

```

```

                                0x46,          /* 70 */
/* 372 */
                                0x12, 0x10,      /* FC_UP [pointer_deref] */
/* 374 */ NdrFcShort( 0x2 ),          /* Offset= 2 (376) */
/* 376 */
                                0x12, 0x0,      /* FC_UP */
/* 378 */ NdrFcShort( 0x1e4 ),        /* Offset= 484 (862) */
/* 380 */
                                0x2a,          /* FC_ENCAPSULATED_UNION */
                                0x89,          /* 137 */
/* 382 */ NdrFcShort( 0x20 ),          /* 32 */
/* 384 */ NdrFcShort( 0xa ),           /* 10 */
/* 386 */ NdrFcLong( 0x8 ),            /* 8 */
/* 390 */ NdrFcShort( 0x50 ),          /* Offset= 80 (470) */
/* 392 */ NdrFcLong( 0xd ),            /* 13 */
/* 396 */ NdrFcShort( 0x70 ),          /* Offset= 112 (508) */
/* 398 */ NdrFcLong( 0x9 ),            /* 9 */
/* 402 */ NdrFcShort( 0x90 ),          /* Offset= 144 (546) */
/* 404 */ NdrFcLong( 0xc ),            /* 12 */
/* 408 */ NdrFcShort( 0xb0 ),          /* Offset= 176 (584) */
/* 410 */ NdrFcLong( 0x24 ),           /* 36 */
/* 414 */ NdrFcShort( 0x102 ),         /* Offset= 258 (672) */
/* 416 */ NdrFcLong( 0x800d ),         /* 32781 */
/* 420 */ NdrFcShort( 0x11e ),         /* Offset= 286 (706) */
/* 422 */ NdrFcLong( 0x10 ),           /* 16 */
/* 426 */ NdrFcShort( 0x138 ),         /* Offset= 312 (738) */
/* 428 */ NdrFcLong( 0x2 ),            /* 2 */
/* 432 */ NdrFcShort( 0x14e ),         /* Offset= 334 (766) */
/* 434 */ NdrFcLong( 0x3 ),            /* 3 */
/* 438 */ NdrFcShort( 0x164 ),         /* Offset= 356 (794) */
/* 440 */ NdrFcLong( 0x14 ),           /* 20 */
/* 444 */ NdrFcShort( 0x17a ),         /* Offset= 378 (822) */
/* 446 */ NdrFcShort( 0xffffffff ),    /* Offset= -1 (445) */
/* 448 */
                                0x21,          /* FC_BOGUS_ARRAY */
                                0x3,          /* 3 */
/* 450 */ NdrFcShort( 0x0 ),           /* 0 */
/* 452 */ 0x19,                        /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 454 */ NdrFcShort( 0x0 ),           /* 0 */
/* 456 */ NdrFcShort( 0x1 ),           /* Corr flags: early, */
/* 458 */ NdrFcLong( 0xffffffff ),     /* -1 */
/* 462 */ NdrFcShort( 0x0 ),           /* Corr flags: */
/* 464 */
                                0x12, 0x0,      /* FC_UP */
/* 466 */ NdrFcShort( 0xffffffff74 ), /* Offset= -140 (326) */
/* 468 */ 0x5c,                        /* FC_PAD */
                                0x5b,          /* FC_END */
/* 470 */
                                0x1a,          /* FC_BOGUS_STRUCT */
                                0x3,          /* 3 */
/* 472 */ NdrFcShort( 0x10 ),          /* 16 */
/* 474 */ NdrFcShort( 0x0 ),           /* 0 */

```

```

/* 476 */      NdrFcShort( 0x6 ),      /* Offset= 6 (482) */
/* 478 */      0x8,      /* FC_LONG */
                0x40,      /* FC_STRUCTPAD4 */
/* 480 */      0x36,      /* FC_POINTER */
                0x5b,      /* FC_END */
/* 482 */
                0x11, 0x0,      /* FC_RP */
/* 484 */      NdrFcShort( 0xffffdc ), /* Offset= -36 (448) */
/* 486 */
                0x21,      /* FC_BOGUS_ARRAY */
                0x3,      /* 3 */
/* 488 */      NdrFcShort( 0x0 ),      /* 0 */
/* 490 */      0x19,      /* Corr desc: field pointer, FC_ULONG */
                0x0,      /* */
/* 492 */      NdrFcShort( 0x0 ),      /* 0 */
/* 494 */      NdrFcShort( 0x1 ),      /* Corr flags: early, */
/* 496 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 500 */      NdrFcShort( 0x0 ),      /* Corr flags: */
/* 502 */      0x4c,      /* FC_EMBEDDED_COMPLEX */
                0x0,      /* 0 */
/* 504 */      NdrFcShort( 0xfffff58 ), /* Offset= -168 (336) */
/* 506 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 508 */
                0x1a,      /* FC_BOGUS_STRUCT */
                0x3,      /* 3 */
/* 510 */      NdrFcShort( 0x10 ),      /* 16 */
/* 512 */      NdrFcShort( 0x0 ),      /* 0 */
/* 514 */      NdrFcShort( 0x6 ),      /* Offset= 6 (520) */
/* 516 */      0x8,      /* FC_LONG */
                0x40,      /* FC_STRUCTPAD4 */
/* 518 */      0x36,      /* FC_POINTER */
                0x5b,      /* FC_END */
/* 520 */
                0x11, 0x0,      /* FC_RP */
/* 522 */      NdrFcShort( 0xffffdc ), /* Offset= -36 (486) */
/* 524 */
                0x21,      /* FC_BOGUS_ARRAY */
                0x3,      /* 3 */
/* 526 */      NdrFcShort( 0x0 ),      /* 0 */
/* 528 */      0x19,      /* Corr desc: field pointer, FC_ULONG */
                0x0,      /* */
/* 530 */      NdrFcShort( 0x0 ),      /* 0 */
/* 532 */      NdrFcShort( 0x1 ),      /* Corr flags: early, */
/* 534 */      NdrFcLong( 0xffffffff ), /* -1 */
/* 538 */      NdrFcShort( 0x0 ),      /* Corr flags: */
/* 540 */      0x4c,      /* FC_EMBEDDED_COMPLEX */
                0x0,      /* 0 */
/* 542 */      NdrFcShort( 0xfffff44 ), /* Offset= -188 (354) */
/* 544 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 546 */
                0x1a,      /* FC_BOGUS_STRUCT */

```

```

0x3, /* 3 */
/* 548 */ NdrFcShort( 0x10 ), /* 16 */
/* 550 */ NdrFcShort( 0x0 ), /* 0 */
/* 552 */ NdrFcShort( 0x6 ), /* Offset= 6 (558) */
/* 554 */ 0x8, /* FC_LONG */
0x40, /* FC_STRUCTPAD4 */
/* 556 */ 0x36, /* FC_POINTER */
0x5b, /* FC_END */
/* 558 */
0x11, 0x0, /* FC_RP */
/* 560 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (524) */
/* 562 */
0x21, /* FC_BOGUS_ARRAY */
0x3, /* 3 */
/* 564 */ NdrFcShort( 0x0 ), /* 0 */
/* 566 */ 0x19, /* Corr desc: field pointer, FC_ULONG */
0x0, /* */
/* 568 */ NdrFcShort( 0x0 ), /* 0 */
/* 570 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 572 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 576 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 578 */
0x12, 0x0, /* FC_UP */
/* 580 */ NdrFcShort( 0x176 ), /* Offset= 374 (954) */
/* 582 */ 0x5c, /* FC_PAD */
0x5b, /* FC_END */
/* 584 */
0x1a, /* FC_BOGUS_STRUCT */
0x3, /* 3 */
/* 586 */ NdrFcShort( 0x10 ), /* 16 */
/* 588 */ NdrFcShort( 0x0 ), /* 0 */
/* 590 */ NdrFcShort( 0x6 ), /* Offset= 6 (596) */
/* 592 */ 0x8, /* FC_LONG */
0x40, /* FC_STRUCTPAD4 */
/* 594 */ 0x36, /* FC_POINTER */
0x5b, /* FC_END */
/* 596 */
0x11, 0x0, /* FC_RP */
/* 598 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (562) */
/* 600 */
0x2f, /* FC_IP */
0x5a, /* FC_CONSTANT_IID */
/* 602 */ NdrFcLong( 0x2f ), /* 47 */
/* 606 */ NdrFcShort( 0x0 ), /* 0 */
/* 608 */ NdrFcShort( 0x0 ), /* 0 */
/* 610 */ 0xc0, /* 192 */
0x0, /* 0 */
/* 612 */ 0x0, /* 0 */
0x0, /* 0 */
/* 614 */ 0x0, /* 0 */
0x0, /* 0 */
/* 616 */ 0x0, /* 0 */
0x46, /* 70 */

```

```

/* 618 */
        0x1b,          /* FC_CARRAY */
        0x0,          /* 0 */
/* 620 */ NdrFcShort( 0x1 ), /* 1 */
/* 622 */ 0x19,        /* Corr desc: field pointer, FC_ULONG */
        0x0,          /* */
/* 624 */ NdrFcShort( 0x4 ), /* 4 */
/* 626 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 628 */ 0x1,        /* FC_BYTE */
        0x5b,        /* FC_END */
/* 630 */
        0x1a,        /* FC_BOGUS_STRUCT */
        0x3,          /* 3 */
/* 632 */ NdrFcShort( 0x18 ), /* 24 */
/* 634 */ NdrFcShort( 0x0 ), /* 0 */
/* 636 */ NdrFcShort( 0xa ), /* Offset= 10 (646) */
/* 638 */ 0x8,        /* FC_LONG */
        0x8,          /* FC_LONG */
/* 640 */ 0x4c,       /* FC_EMBEDDED_COMPLEX */
        0x0,          /* 0 */
/* 642 */ NdrFcShort( 0xfffffd6 ), /* Offset= -42 (600) */
/* 644 */ 0x36,       /* FC_POINTER */
        0x5b,        /* FC_END */
/* 646 */
        0x12, 0x0,    /* FC_UP */
/* 648 */ NdrFcShort( 0xfffffe2 ), /* Offset= -30 (618) */
/* 650 */
        0x21,        /* FC_BOGUS_ARRAY */
        0x3,          /* 3 */
/* 652 */ NdrFcShort( 0x0 ), /* 0 */
/* 654 */ 0x19,       /* Corr desc: field pointer, FC_ULONG */
        0x0,          /* */
/* 656 */ NdrFcShort( 0x0 ), /* 0 */
/* 658 */ NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 660 */ NdrFcLong( 0xffffffff ), /* -1 */
/* 664 */ NdrFcShort( 0x0 ), /* Corr flags: */
/* 666 */
        0x12, 0x0,    /* FC_UP */
/* 668 */ NdrFcShort( 0xfffffda ), /* Offset= -38 (630) */
/* 670 */ 0x5c,       /* FC_PAD */
        0x5b,        /* FC_END */
/* 672 */
        0x1a,        /* FC_BOGUS_STRUCT */
        0x3,          /* 3 */
/* 674 */ NdrFcShort( 0x10 ), /* 16 */
/* 676 */ NdrFcShort( 0x0 ), /* 0 */
/* 678 */ NdrFcShort( 0x6 ), /* Offset= 6 (684) */
/* 680 */ 0x8,        /* FC_LONG */
        0x40,        /* FC_STRUCTPAD4 */
/* 682 */ 0x36,       /* FC_POINTER */
        0x5b,        /* FC_END */
/* 684 */
        0x11, 0x0,    /* FC_RP */

```

```

/* 686 */      NdrFcShort( 0xfffffdc ), /* Offset= -36 (650) */
/* 688 */
                0x1d,          /* FC_SMFARRAY */
                0x0,          /* 0 */
/* 690 */      NdrFcShort( 0x8 ), /* 8 */
/* 692 */      0x1,          /* FC_BYTE */
                0x5b,          /* FC_END */
/* 694 */
                0x15,          /* FC_STRUCT */
                0x3,          /* 3 */
/* 696 */      NdrFcShort( 0x10 ), /* 16 */
/* 698 */      0x8,          /* FC_LONG */
                0x6,          /* FC_SHORT */
/* 700 */      0x6,          /* FC_SHORT */
                0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 702 */      0x0,          /* 0 */
                NdrFcShort( 0xfffff1 ), /* Offset= -15 (688) */
                0x5b,          /* FC_END */
/* 706 */
                0x1a,          /* FC_BOGUS_STRUCT */
                0x3,          /* 3 */
/* 708 */      NdrFcShort( 0x20 ), /* 32 */
/* 710 */      NdrFcShort( 0x0 ), /* 0 */
/* 712 */      NdrFcShort( 0xa ), /* Offset= 10 (722) */
/* 714 */      0x8,          /* FC_LONG */
                0x40,          /* FC_STRUCTPAD4 */
/* 716 */      0x36,          /* FC_POINTER */
                0x4c,          /* FC_EMBEDDED_COMPLEX */
/* 718 */      0x0,          /* 0 */
                NdrFcShort( 0xfffffe7 ), /* Offset= -25 (694) */
                0x5b,          /* FC_END */
/* 722 */
                0x11, 0x0,      /* FC_RP */
/* 724 */      NdrFcShort( 0xfffff12 ), /* Offset= -238 (486) */
/* 726 */
                0x1b,          /* FC_CARRAY */
                0x0,          /* 0 */
/* 728 */      NdrFcShort( 0x1 ), /* 1 */
/* 730 */      0x19,          /* Corr desc: field pointer, FC_ULONG */
                0x0,          /* */
/* 732 */      NdrFcShort( 0x0 ), /* 0 */
/* 734 */      NdrFcShort( 0x1 ), /* Corr flags: early, */
/* 736 */      0x1,          /* FC_BYTE */
                0x5b,          /* FC_END */
/* 738 */
                0x1a,          /* FC_BOGUS_STRUCT */
                0x3,          /* 3 */
/* 740 */      NdrFcShort( 0x10 ), /* 16 */
/* 742 */      NdrFcShort( 0x0 ), /* 0 */
/* 744 */      NdrFcShort( 0x6 ), /* Offset= 6 (750) */
/* 746 */      0x8,          /* FC_LONG */
                0x40,          /* FC_STRUCTPAD4 */
/* 748 */      0x36,          /* FC_POINTER */

```



```

                                0x5b,          /* FC_END */
/* 750 */
                                0x12, 0x0,      /* FC_UP */
/* 752 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (726) */
/* 754 */
                                0x1b,          /* FC_CARRAY */
                                0x1,          /* 1 */
/* 756 */ NdrFcShort( 0x2 ),          /* 2 */
/* 758 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 760 */ NdrFcShort( 0x0 ),          /* 0 */
/* 762 */ NdrFcShort( 0x1 ),          /* Corr flags: early, */
/* 764 */ 0x6,          /* FC_SHORT */
                                0x5b,          /* FC_END */
/* 766 */
                                0x1a,          /* FC_BOGUS_STRUCT */
                                0x3,          /* 3 */
/* 768 */ NdrFcShort( 0x10 ),          /* 16 */
/* 770 */ NdrFcShort( 0x0 ),          /* 0 */
/* 772 */ NdrFcShort( 0x6 ),          /* Offset= 6 (778) */
/* 774 */ 0x8,          /* FC_LONG */
                                0x40,          /* FC_STRUCTPAD4 */
/* 776 */ 0x36,          /* FC_POINTER */
                                0x5b,          /* FC_END */
/* 778 */
                                0x12, 0x0,      /* FC_UP */
/* 780 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (754) */
/* 782 */
                                0x1b,          /* FC_CARRAY */
                                0x3,          /* 3 */
/* 784 */ NdrFcShort( 0x4 ),          /* 4 */
/* 786 */ 0x19,          /* Corr desc: field pointer, FC_ULONG */
                                0x0,          /* */
/* 788 */ NdrFcShort( 0x0 ),          /* 0 */
/* 790 */ NdrFcShort( 0x1 ),          /* Corr flags: early, */
/* 792 */ 0x8,          /* FC_LONG */
                                0x5b,          /* FC_END */
/* 794 */
                                0x1a,          /* FC_BOGUS_STRUCT */
                                0x3,          /* 3 */
/* 796 */ NdrFcShort( 0x10 ),          /* 16 */
/* 798 */ NdrFcShort( 0x0 ),          /* 0 */
/* 800 */ NdrFcShort( 0x6 ),          /* Offset= 6 (806) */
/* 802 */ 0x8,          /* FC_LONG */
                                0x40,          /* FC_STRUCTPAD4 */
/* 804 */ 0x36,          /* FC_POINTER */
                                0x5b,          /* FC_END */
/* 806 */
                                0x12, 0x0,      /* FC_UP */
/* 808 */ NdrFcShort( 0xfffffe6 ), /* Offset= -26 (782) */
/* 810 */
                                0x1b,          /* FC_CARRAY */
                                0x7,          /* 7 */

```

```

/* 812 */      NdrFcShort( 0x8 ),      /* 8 */
/* 814 */      0x19,      /* Corr desc: field pointer, FC_ULONG */
                0x0,      /* */
/* 816 */      NdrFcShort( 0x0 ),      /* 0 */
/* 818 */      NdrFcShort( 0x1 ),      /* Corr flags: early, */
/* 820 */      0xb,      /* FC_HYPER */
                0x5b,      /* FC_END */
/* 822 */
                0x1a,      /* FC_BOGUS_STRUCT */
                0x3,      /* 3 */
/* 824 */      NdrFcShort( 0x10 ),      /* 16 */
/* 826 */      NdrFcShort( 0x0 ),      /* 0 */
/* 828 */      NdrFcShort( 0x6 ),      /* Offset= 6 (834) */
/* 830 */      0x8,      /* FC_LONG */
                0x40,      /* FC_STRUCTPAD4 */
/* 832 */      0x36,      /* FC_POINTER */
                0x5b,      /* FC_END */
/* 834 */
                0x12, 0x0,      /* FC_UP */
/* 836 */      NdrFcShort( 0xfffffe6 ), /* Offset= -26 (810) */
/* 838 */
                0x15,      /* FC_STRUCT */
                0x3,      /* 3 */
/* 840 */      NdrFcShort( 0x8 ),      /* 8 */
/* 842 */      0x8,      /* FC_LONG */
                0x8,      /* FC_LONG */
/* 844 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 846 */
                0x1b,      /* FC_CARRAY */
                0x3,      /* 3 */
/* 848 */      NdrFcShort( 0x8 ),      /* 8 */
/* 850 */      0x7,      /* Corr desc: FC_USHORT */
                0x0,      /* */
/* 852 */      NdrFcShort( 0xffc8 ),      /* -56 */
/* 854 */      NdrFcShort( 0x1 ),      /* Corr flags: early, */
/* 856 */      0x4c,      /* FC_EMBEDDED_COMPLEX */
                0x0,      /* 0 */
/* 858 */      NdrFcShort( 0xfffffec ), /* Offset= -20 (838) */
/* 860 */      0x5c,      /* FC_PAD */
                0x5b,      /* FC_END */
/* 862 */
                0x1a,      /* FC_BOGUS_STRUCT */
                0x3,      /* 3 */
/* 864 */      NdrFcShort( 0x38 ),      /* 56 */
/* 866 */      NdrFcShort( 0xfffffec ), /* Offset= -20 (846) */
/* 868 */      NdrFcShort( 0x0 ),      /* Offset= 0 (868) */
/* 870 */      0x6,      /* FC_SHORT */
                0x6,      /* FC_SHORT */
/* 872 */      0x8,      /* FC_LONG */
                0x8,      /* FC_LONG */
/* 874 */      0x40,      /* FC_STRUCTPAD4 */
                0x4c,      /* FC_EMBEDDED_COMPLEX */

```

```

/* 876 */      0x0,          /* 0 */
                NdrFcShort( 0xffffe0f ), /* Offset= -497 (380) */
                0x5b,          /* FC_END */

/* 880 */
                0x12, 0x0,      /* FC_UP */
/* 882 */      NdrFcShort( 0xfffff04 ), /* Offset= -252 (630) */
/* 884 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 886 */      0x1,          /* FC_BYTE */
                0x5c,          /* FC_PAD */

/* 888 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 890 */      0x6,          /* FC_SHORT */
                0x5c,          /* FC_PAD */

/* 892 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 894 */      0x8,          /* FC_LONG */
                0x5c,          /* FC_PAD */

/* 896 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 898 */      0xb,          /* FC_HYPER */
                0x5c,          /* FC_PAD */

/* 900 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 902 */      0xa,          /* FC_FLOAT */
                0x5c,          /* FC_PAD */

/* 904 */
                0x12, 0x8,      /* FC_UP [simple_pointer] */
/* 906 */      0xc,          /* FC_DOUBLE */
                0x5c,          /* FC_PAD */

/* 908 */
                0x12, 0x0,      /* FC_UP */
/* 910 */      NdrFcShort( 0xffffda2 ), /* Offset= -606 (304) */
/* 912 */
                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 914 */      NdrFcShort( 0xffffda4 ), /* Offset= -604 (310) */
/* 916 */
                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 918 */      NdrFcShort( 0xffffdba ), /* Offset= -582 (336) */
/* 920 */
                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 922 */      NdrFcShort( 0xffffdc8 ), /* Offset= -568 (354) */
/* 924 */
                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 926 */      NdrFcShort( 0xffffdd6 ), /* Offset= -554 (372) */
/* 928 */
                0x12, 0x10,     /* FC_UP [pointer_deref] */
/* 930 */      NdrFcShort( 0x2 ), /* Offset= 2 (932) */
/* 932 */
                0x12, 0x0,      /* FC_UP */
/* 934 */      NdrFcShort( 0x14 ), /* Offset= 20 (954) */
/* 936 */
                0x15,          /* FC_STRUCT */

```

```

        0x7,          /* 7 */
/* 938 */ NdrFcShort( 0x10 ), /* 16 */
/* 940 */ 0x6,          /* FC_SHORT */
        0x1,          /* FC_BYTE */
/* 942 */ 0x1,          /* FC_BYTE */
        0x8,          /* FC_LONG */
/* 944 */ 0xb,          /* FC_HYPER */
        0x5b,         /* FC_END */
/* 946 */
        0x12, 0x0,     /* FC_UP */
/* 948 */ NdrFcShort( 0xffffffff4 ), /* Offset= -12 (936) */
/* 950 */
        0x12, 0x8,     /* FC_UP [simple_pointer] */
/* 952 */ 0x2,          /* FC_CHAR */
        0x5c,         /* FC_PAD */
/* 954 */
        0x1a,          /* FC_BOGUS_STRUCT */
        0x7,          /* 7 */
/* 956 */ NdrFcShort( 0x20 ), /* 32 */
/* 958 */ NdrFcShort( 0x0 ), /* 0 */
/* 960 */ NdrFcShort( 0x0 ), /* Offset= 0 (960) */
/* 962 */ 0x8,          /* FC_LONG */
        0x8,          /* FC_LONG */
/* 964 */ 0x6,          /* FC_SHORT */
        0x6,          /* FC_SHORT */
/* 966 */ 0x6,          /* FC_SHORT */
        0x6,          /* FC_SHORT */
/* 968 */ 0x4c,         /* FC_EMBEDDED_COMPLEX */
        0x0,          /* 0 */
/* 970 */ NdrFcShort( 0xffffc3c ), /* Offset= -964 (6) */
/* 972 */ 0x5c,         /* FC_PAD */
        0x5b,         /* FC_END */
/* 974 */ 0xb4,         /* FC_USER_MARSHAL */
        0x83,         /* 131 */
/* 976 */ NdrFcShort( 0x0 ), /* 0 */
/* 978 */ NdrFcShort( 0x18 ), /* 24 */
/* 980 */ NdrFcShort( 0x0 ), /* 0 */
/* 982 */ NdrFcShort( 0xffffc2c ), /* Offset= -980 (2) */
/* 984 */
        0x11, 0x4,     /* FC_RP [allocated_on_stack] */
/* 986 */ NdrFcShort( 0x6 ), /* Offset= 6 (992) */
/* 988 */
        0x13, 0x0,     /* FC_OP */
/* 990 */ NdrFcShort( 0xfffffdc ), /* Offset= -36 (954) */
/* 992 */ 0xb4,         /* FC_USER_MARSHAL */
        0x83,         /* 131 */
/* 994 */ NdrFcShort( 0x0 ), /* 0 */
/* 996 */ NdrFcShort( 0x18 ), /* 24 */
/* 998 */ NdrFcShort( 0x0 ), /* 0 */
/* 1000 */ NdrFcShort( 0xffffffff4 ), /* Offset= -12 (988) */

        0x0
    }

```

```

};

static const USER_MARSHAL_ROUTINE_QUADRUPLE
UserMarshalRoutines[ WIRE_MARSHAL_TABLE_SIZE ] =
{
    {
        VARIANT_UserSize
        ,VARIANT_UserMarshal
        ,VARIANT_UserUnmarshal
        ,VARIANT_UserFree
    }
};

```

```

/* Standard interface: __MIDL_itf_tpcc_com_ps_0000, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}} */

```

```

/* Object interface: IUnknown, ver. 0.0,
   GUID={0x00000000,0x0000,0x0000,{0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46}} */

```

```

/* Object interface: ITPCC, ver. 0.0,
   GUID={0xFEEE6AA2,0x84B1,0x11d2,{0xBA,0x47,0x00,0xC0,0x4F,0xBF,0xE0,0x8B}} */

```

```

#pragma code_seg(".orpc")
static const unsigned short ITPCC_FormatStringOffsetTable[] =
{
    0,
    44,
    88,
    132,
    176,
    220
};

```

```

static const MIDL_STUBLESS_PROXY_INFO ITPCC_ProxyInfo =
{
    &Object_StubDesc,
    __MIDL_ProcFormatString.Format,
    &ITPCC_FormatStringOffsetTable[-3],
    0,
    0,
    0
};

```

```

static const MIDL_SERVER_INFO ITPCC_ServerInfo =
{
    &Object_StubDesc,

```

```

0,
__MIDL_ProcFormatString.Format,
&ITPCC_FormatStringOffsetTable[-3],
0,
0,
0,
0};
CINTERFACE_PROXY_VTABLE(9) _ITPCCProxyVtbl =
{
    &ITPCC_ProxyInfo,
    &IID_ITPCC,
    IUnknown_QueryInterface_Proxy,
    IUnknown_AddRef_Proxy,
    IUnknown_Release_Proxy,
    (void *) (INT_PTR) -1 /* ITPCC::NewOrder */,
    (void *) (INT_PTR) -1 /* ITPCC::Payment */,
    (void *) (INT_PTR) -1 /* ITPCC::Delivery */,
    (void *) (INT_PTR) -1 /* ITPCC::StockLevel */,
    (void *) (INT_PTR) -1 /* ITPCC::OrderStatus */,
    (void *) (INT_PTR) -1 /* ITPCC::CallSetComplete */
};

const CInterfaceStubVtbl _ITPCCStubVtbl =
{
    &IID_ITPCC,
    &ITPCC_ServerInfo,
    9,
    0, /* pure interpreted */
    CStdStubBuffer_METHODS
};

static const MIDL_STUB_DESC Object_StubDesc =
{
    0,
    NdrOleAllocate,
    NdrOleFree,
    0,
    0,
    0,
    0,
    0,
    0,
    __MIDL_TypeFormatString.Format,
    1, /* -error bounds_check flag */
    0x50002, /* Ndr library version */
    0,
    0x600015b, /* MIDL Version 6.0.347 */
    0,
    UserMarshalRoutines,
    0, /* notify & notify_flag routine table */
    0x1, /* MIDL flag */
    0, /* cs routines */
    0, /* proxy/server info */
    0 /* Reserved5 */
};

```

```

};

const CInterfaceProxyVtbl * _tpcc_com_ps_ProxyVtblList[] =
{
    ( CInterfaceProxyVtbl *) &_ITPCCProxyVtbl,
    0
};

const CInterfaceStubVtbl * _tpcc_com_ps_StubVtblList[] =
{
    ( CInterfaceStubVtbl *) &_ITPCCStubVtbl,
    0
};

PCInterfaceName const _tpcc_com_ps_InterfaceNamesList[] =
{
    "ITPCC",
    0
};

#define _tpcc_com_ps_CHECK_IID(n) IID_GENERIC_CHECK_IID( _tpcc_com_ps, pIID, n)

int __stdcall _tpcc_com_ps_IID_Lookup( const IID * pIID, int * pIndex )
{
    if(!_tpcc_com_ps_CHECK_IID(0))
    {
        *pIndex = 0;
        return 1;
    }

    return 0;
}

const ExtendedProxyFileInfo tpcc_com_ps_ProxyFileInfo =
{
    (PCInterfaceProxyVtblList *) &_tpcc_com_ps_ProxyVtblList,
    (PCInterfaceStubVtblList *) &_tpcc_com_ps_StubVtblList,
    (const PCInterfaceName * ) &_tpcc_com_ps_InterfaceNamesList,
    0, // no delegation
    &_tpcc_com_ps_IID_Lookup,
    1,
    2,
    0, /* table of [async_uuid] interfaces */
    0, /* Filler1 */
    0, /* Filler2 */
    0 /* Filler3 */
};

#endif /* defined(_M_IA64) || defined(_M_AMD64)*/

```

tuxapp.cpp

```
/*      FILE:          TUXAPP.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 server.
 *      Contact: Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *      4.20.000 - updated rev number to match kit
 */

#include <windows.h>
#include <process.h>
#include <tchar.h>
#include <stdio.h>
#include <stdarg.h>
#include <iostream.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>
#include <sql.h>
#include <sqlext.h>

#include <tmenv.h>
#include <xa.h>
#include <atmi.h>

#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 "..\..\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 "tuxapp.h"

char    szMyComputerName[MAX_COMPUTERNAME_LENGTH+1];

// configuration settings from registry
```



```

TPCCREGISTRYDATA      Reg;

CTPCC_BASE            *pTxn = NULL;

#include "..\..\common\src\ReadRegistry.cpp"

/* FUNCTION: tpsvrinit ( int argc, char *argv[] )
 *
 * PURPOSE:   Initialize the Server to Database connection.
 *
 * RETURNS:   int      0      Success
 *            int     -1      Failure
 */

int tpsvrinit ( int argc, char *argv[] )
{
    try
    {
        DWORD dwSize = MAX_COMPUTERNAME_LENGTH+1;
        GetComputerName(szMyComputerName, &dwSize);
        szMyComputerName[dwSize] = 0;

        if ( ReadTPCCRegistrySettings( &Reg ) )
            throw new CTUXAPP_ERR( ERR_MISSING_REGISTRY_ENTRIES );

        GetParameters(argc, argv);

        switch (Reg.eDB_Protocol)
        {
            case ODBC:
                pTxn = new CTPCC_ODBC( Reg.szDbServer, Reg.szDbUser,
                Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                break;
            case DBLIB:
                pTxn = new CTPCC_DBLIB( Reg.szDbServer, Reg.szDbUser,
                Reg.szDbPassword, szMyComputerName, Reg.szDbName );
                break;
        }
    }
    catch (CBaseErr *e)
    {
        WriteMessageToEventLog(e->ErrorText());
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception."));
    }

    return 0;
}

```

```

/* FUNCTION: tpsvrdone ( void )
*
*/

void tpsvrdone ( void )
{
    delete pTxn;
    pTxn = NULL;
}

/* FUNCTION: BOOL GetParameters(int argc, char *argv[])
*
* PURPOSE: This function parses the command line passed in to the delivery executable,
initializing
*
*           and filling in global variable parameters.
*
* ARGUMENTS: int      argc    number of command line arguments passed to
delivery
*
*           char    *argv[]  array of command line argument pointers
*
*/

static void GetParameters(int argc, char *argv[])
{
    // advance through args until "--" is found
    for(int j=0; j<argc; j++)
    {
        if (strcmp(argv[j],"--") == 0)
            break;
    }

    for(int i=j+1; i<argc; i++)
    {
        if ( argv[i][0] == '-' || argv[i][0] == '/' )
        {
            switch(argv[i][1])
            {
                case 'S':
                    strcpy(Reg.szDbServer, argv[i]+2);
                    break;
                case 'D':
                    strcpy(Reg.szDbName, argv[i]+2);
                    break;
                case 'P':
                    strcpy(Reg.szDbPassword, argv[i]+2);
                    break;
                case 'U':
                    strcpy(Reg.szDbUser, argv[i]+2);
                    break;
                default:
                    cout << "Microsoft TPC-C Kit" << endl;
                    cout << "Tuxedo Server" << endl << endl;
                    cout << "Usage:" << endl;
            }
        }
    }
}

```

```

        cout << "    tuxapp [<tuxedo-args>] -- -S<sql-server> [-
D<database>] [-U<user>] [-P<password>]" << endl << endl;
        cout << "All parameters default to values in registry." <<
endl;
        throw new CTUXAPP_ERR( ERR_BAD_SYNTAX );
    }
}
}
}
}

```

```

static void WriteMessageToEventLog(LPTSTR lpszMsg)
{
    TCHAR  szMsg[256];
    HANDLE hEventSource;
    LPTSTR  lpszStrings[2];

    // Use event logging to log the error.
    //
    hEventSource = RegisterEventSource(NULL, TEXT("TUXAPP.EXE"));

    _stprintf(szMsg, TEXT("Error in TUXAPP.EXE: "));
    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);
    }
}
}
}
}
}

```

```

void NEWORDER( TPSVCINFO *rqst )
{
    PNEW_ORDER_DATA pNewOrder;
    TUX_DATA *pData;
    const int iSize = sizeof(pData->u.NewOrder);

    try
    {
        pData = (TUX_DATA *)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pNewOrder = pTxn->BuffAddr_NewOrder();
    }
}
}
}
}
}

```

```

        assert( rqst->len == sizeof(TUX_DATA) );
        memcpy(pNewOrder, &pData->u.NewOrder, iSize );

        pTxn->NewOrder();
        memcpy( &pData->u.NewOrder, pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.NewOrder, pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.NewOrder, pNewOrder, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
}

void PAYMENT( TPSVCINFO *rqst )
{
    PPAYMENT_DATA    pPayment;
    TUX_DATA          *pData;
    const int         iSize = sizeof(pData->u.Payment);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pPayment = pTxn->BuffAddr_Payment();
        assert( rqst->len == sizeof(TUX_DATA) );
        memcpy(pPayment, &pData->u.Payment, iSize );

        pTxn->Payment();
        memcpy( &pData->u.Payment, pPayment, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.Payment, pPayment, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
        delete e;
    }
}

```

```

catch (...)
{
    WriteMessageToEventLog(TEXT("Unhandled exception.));
    pData->retval = ERR_TYPE_LOGIC;
    pData->error = 0;
    memcpy( &pData->u.Payment, pPayment, iSize );
    tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
}
}

```

// Note: Delivery txn code below does not implement logging of the delivery
// txn results, so cannot be used as is to run an auditable TPC-C result.
// The code is included for completeness.

```

void DELIVERY( TPSVCINFO *rqst )
{
    PDELIVERY_DATA    pDelivery;
    TUX_DATA          *pData;
    const int         iSize = sizeof(pData->u.Delivery);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pDelivery = pTxn->BuffAddr_Delivery();
        assert( rqst->len == sizeof(TUX_DATA) );
        memcpy(pDelivery, &pData->u.Delivery, iSize );

        pTxn->Delivery();

        memcpy( &pData->u.Delivery, pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.Delivery, pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.Delivery, pDelivery, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
}

```

```

void STOCKLEVEL( TPSVCINFO *rqst )

```

```

{
    PSTOCK_LEVEL_DATA    pStockLevel;
    TUX_DATA              *pData;
    const int             iSize = sizeof(pData->u.StockLevel);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pStockLevel = pTxn->BuffAddr_StockLevel();
        assert( rqst->len == sizeof(TUX_DATA) );
        memcpy(pStockLevel, &pData->u.StockLevel, iSize );

        pTxn->StockLevel();
        memcpy( &pData->u.StockLevel, pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.StockLevel, pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception. "));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.StockLevel, pStockLevel, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
}

```

```

void ORDERSTATUS( TPSVCINFO *rqst )
{
    PORDER_STATUS_DATA    pOrderStatus;
    TUX_DATA              *pData;
    const int iSize = sizeof(pData->u.OrderStatus);

    try
    {
        pData = (TUX_DATA*)rqst->data;
        pData->retval = ERR_SUCCESS;
        pData->error = 0;

        pOrderStatus = pTxn->BuffAddr_OrderStatus();
        assert( rqst->len == sizeof(TUX_DATA) );
        memcpy(pOrderStatus, &pData->u.OrderStatus, iSize );
    }
}

```

```

        pTxn->OrderStatus();
        memcpy( &pData->u.OrderStatus, pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
    catch (CBaseErr *e)
    {
        pData->retval = e->ErrorType();
        pData->error = e->ErrorNum();
        memcpy( &pData->u.OrderStatus, pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
        delete e;
    }
    catch (...)
    {
        WriteMessageToEventLog(TEXT("Unhandled exception.));
        pData->retval = ERR_TYPE_LOGIC;
        pData->error = 0;
        memcpy( &pData->u.OrderStatus, pOrderStatus, iSize );
        tpreturn( TPSUCCESS, 0, rqst->data, sizeof(TUX_DATA), 0);
    }
}

/* FUNCTION: CTUXAPP_ERR::ErrorText
 *
 */

char* CTUXAPP_ERR::ErrorText(void)
{
    int i;

    static SERRORMSG errorMsgs[] =
    {
        { ERR_MISSING_REGISTRY_ENTRIES,      "Required entries missing from
registry." },
        { ERR_BAD_SYNTAX,                    "Syntax error in input
parameters." },
        { ERR_UNKNOWN_DB_PROTOCOL,          "Unknown database protocol
specified in registry." },
        { 0,                                  "" }
    };

    static char szNotFound[] = "Unknown error number.";

    for(i=0; errorMsgs[i].szMsg[0]; i++)
    {
        if ( m_Error == errorMsgs[i].iError )
            break;
    }
    if ( !errorMsgs[i].szMsg[0] )
        return szNotFound;
    else
        return errorMsgs[i].szMsg;
}

```

```
}
```

tuxapp.h

```
/* FILE: TUXAPP.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 Tuxedo server.
 *
 * Change history:
 * 4.20.000 - updated rev number to match kit
 */
```

```
enum TUXERROR
{
    ERR_MISSING_REGISTRY_ENTRIES = 1,
    ERR_BAD_SYNTAX,
    ERR_UNKNOWN_DB_PROTOCOL
};
```

```
class CTUXAPP_ERR : public CBaseErr
{
public:
    TUXERROR m_Error;

    CTUXAPP_ERR(TUXERROR Err) { m_Error = Err; };
    ~CTUXAPP_ERR() {};

    int ErrorType() {return ERR_TYPE_TUXEDO;};
    int ErrorNum() {return m_Error;};
    char *ErrorText();
};
```

```
struct TUX_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;
};

static void GetParameters(int argc, char *argv[]);
static void WriteMessageToEventLog(LPTSTR lpszMsg);

#ifdef __cplusplus
extern "C" {
#endif

void NEWORDER( TPSVCINFO *rqst );
void PAYMENT( TPSVCINFO *rqst );
void DELIVERY( TPSVCINFO *rqst );
void STOCKLEVEL( TPSVCINFO *rqst );
void ORDERSTATUS( TPSVCINFO *rqst );

#ifdef __cplusplus
}
#endif

```

tuxmain.c

```

/*      FILE:          TUXMAIN.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:      Implementation for TPC-C Tuxedo server.
 *      Contact: Charles Levine (clevine@microsoft.com)
 *
 *      Change history:
 *                  4.20.000 - updated rev number to match kit
 */

#include <stdio.h>
#include <xa.h>
#include <atmi.h>

#ifdef __cplusplus
extern "C" {
#endif
extern int _tmrunserver _((int));
extern void DELIVERY _((TPSVCINFO *));
extern void NEWORDER _((TPSVCINFO *));
extern void ORDERSTATUS _((TPSVCINFO *));
extern void PAYMENT _((TPSVCINFO *));
extern void STOCKLEVEL _((TPSVCINFO *));

```

```

#if defined(__cplusplus)
}
#endif

static struct tmdsptchtbl_t _tmdsptchtbl[] = {
    { "DELIVERY", "DELIVERY", (void (*)_((TPSVCINFO *))) DELIVERY, 0, 0 },
    { "NEWORDER", "NEWORDER", (void (*)_((TPSVCINFO *))) NEWORDER, 1, 0 },
    { "ORDERSTATUS", "ORDERSTATUS", (void (*)_((TPSVCINFO *))) ORDERSTATUS, 2,
0 },
    { "PAYMENT", "PAYMENT", (void (*)_((TPSVCINFO *))) PAYMENT, 3, 0 },
    { "STOCKLEVEL", "STOCKLEVEL", (void (*)_((TPSVCINFO *))) STOCKLEVEL, 4,
0 },
    { NULL, NULL, NULL, 0, 0 }
};

#ifdef _TMDLLIMPORT
#define _TMDLLIMPORT
#endif

_TMDLLIMPORT extern struct xa_switch_t tmnull_switch;

struct tmsvrargs_t tmsvrargs = {
    NULL,
    &_tmdsptchtbl[0],
    0,
    tpsvrinit,
    tpsvrdone,
    _tmrunserver, /* PRIVATE */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL, /* RESERVED */
    NULL /* RESERVED */
};

struct tmsvrargs_t *
#ifdef _TMPROTOTYPES
_tmgetsvrargs(void)
#else
_tmgetsvrargs()
#endif
{
    tmsvrargs.xa_switch = &tmnull_switch;
    return(&tmsvrargs);
}

int
#ifdef _TMPROTOTYPES
main(int argc, char **argv)
#else
main(argc,argv)
int argc;
char **argv;
#endif

```

```

{
#ifdef TMMAINEXIT
#include "mainexit.h"
#endif

    return( _tmstartserver( argc, argv, _tmgetsvrargs()));
}

```

WebcInt.sln

Microsoft Visual Studio Solution File, Format Version 7.00

```

Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "db_dblib_dll",
"db_dblib_dll\db_dblib_dll.vcproj", "{90B8CE0A-384C-4ACD-8D75-2514A8320111}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "db_odbc_dll",
"db_odbc_dll\db_odbc_dll.vcproj", "{ED60E841-5271-4AFB-9380-3C2B30DE72A2}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "install", "install\install.vcproj",
"{5B1F940A-453D-4A87-A31B-EB058641F45B}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "isapi_dll",
"isapi_dll\isapi_dll.vcproj", "{537A91EB-A43E-4E7F-991C-69C101208B27}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "tm_com_dll",
"tm_com_dll\tm_com_dll.vcproj", "{0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "tpcc_com_all",
"tpcc_com_all\tpcc_com_all.vcproj", "{7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "tpcc_com_ps",
"tpcc_com_ps\tpcc_com_ps.vcproj", "{7E895AFF-A266-4AB4-B37F-E0189653C944}"
EndProject
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "db_oledb_tpcc",
"db_oledb_dll\db_oledb_tpcc.vcproj", "{FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}"
EndProject
Global
    GlobalSection(SolutionConfiguration) = preSolution
        ConfigName.0 = Debug
        ConfigName.1 = IceCAP
        ConfigName.2 = Release
    EndGlobalSection
    GlobalSection(ProjectDependencies) = postSolution
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.0 = {7E895AFF-A266-4AB4-
B37F-E0189653C944}
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.1 = {7F0B19EC-B9E9-4D9F-
97B2-6F4935ECD617}
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.2 = {0DD8F08B-6CD3-45BE-
8189-80BDCF22A22C}
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.3 = {ED60E841-5271-4AFB-
9380-3C2B30DE72A2}
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.4 = {90B8CE0A-384C-4ACD-

```

```

8D75-2514A8320111}
    {5B1F940A-453D-4A87-A31B-EB058641F45B}.5 = {537A91EB-A43E-4E7F-
991C-69C101208B27}
    {537A91EB-A43E-4E7F-991C-69C101208B27}.0 = {0DD8F08B-6CD3-45BE-
8189-80BDCF22A22C}
    {537A91EB-A43E-4E7F-991C-69C101208B27}.1 = {ED60E841-5271-4AFB-
9380-3C2B30DE72A2}
    {537A91EB-A43E-4E7F-991C-69C101208B27}.2 = {90B8CE0A-384C-4ACD-
8D75-2514A8320111}
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.0 = {7F0B19EC-B9E9-4D9F-
97B2-6F4935ECD617}
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.1 = {7E895AFF-A266-4AB4-
B37F-E0189653C944}
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.0 = {7E895AFF-A266-4AB4-
B37F-E0189653C944}
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.1 = {90B8CE0A-384C-4ACD-
8D75-2514A8320111}
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.2 = {ED60E841-5271-4AFB-
9380-3C2B30DE72A2}
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.3 = {FA0E8F97-BCE7-4DDD-
A6AE-DD53F31DC960}
    EndGlobalSection
    GlobalSection(ProjectConfiguration) = postSolution
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.Debug.ActiveCfg = Debug|Win32
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.Debug.Build.0 = Debug|Win32
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.IceCAP.ActiveCfg =
IceCAP|Win32
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.IceCAP.Build.0 = IceCAP|Win32
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.Release.ActiveCfg =
Release|Win32
        {90B8CE0A-384C-4ACD-8D75-2514A8320111}.Release.Build.0 = Release|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.Debug.ActiveCfg = Debug|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.Debug.Build.0 = Debug|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.IceCAP.ActiveCfg =
IceCAP|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.IceCAP.Build.0 = IceCAP|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.Release.ActiveCfg =
Release|Win32
        {ED60E841-5271-4AFB-9380-3C2B30DE72A2}.Release.Build.0 = Release|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.Debug.ActiveCfg = Debug|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.Debug.Build.0 = Debug|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.IceCAP.ActiveCfg =
Debug|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.IceCAP.Build.0 = Debug|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.Release.ActiveCfg =
Release|Win32
        {5B1F940A-453D-4A87-A31B-EB058641F45B}.Release.Build.0 = Release|Win32
        {537A91EB-A43E-4E7F-991C-69C101208B27}.Debug.ActiveCfg = Debug|Win32
        {537A91EB-A43E-4E7F-991C-69C101208B27}.Debug.Build.0 = Debug|Win32
        {537A91EB-A43E-4E7F-991C-69C101208B27}.IceCAP.ActiveCfg =
IceCAP|Win32
        {537A91EB-A43E-4E7F-991C-69C101208B27}.IceCAP.Build.0 = IceCAP|Win32
        {537A91EB-A43E-4E7F-991C-69C101208B27}.Release.ActiveCfg =

```

```

Release|Win32
    {537A91EB-A43E-4E7F-991C-69C101208B27}.Release.Build.0 = Release|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.Debug.ActiveCfg =
Debug|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.Debug.Build.0 = Debug|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.IceCAP.ActiveCfg =
Release|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.IceCAP.Build.0 = Release|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.Release.ActiveCfg =
Release|Win32
    {0DD8F08B-6CD3-45BE-8189-80BDCF22A22C}.Release.Build.0 = Release|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.Debug.ActiveCfg = Debug|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.Debug.Build.0 = Debug|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.IceCAP.ActiveCfg =
Debug|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.IceCAP.Build.0 = Debug|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.Release.ActiveCfg =
Release|Win32
    {7F0B19EC-B9E9-4D9F-97B2-6F4935ECD617}.Release.Build.0 = Release|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.Debug.ActiveCfg = Debug|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.Debug.Build.0 = Debug|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.IceCAP.ActiveCfg =
Release|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.IceCAP.Build.0 = Release|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.Release.ActiveCfg =
Release|Win32
    {7E895AFF-A266-4AB4-B37F-E0189653C944}.Release.Build.0 = Release|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.Debug.ActiveCfg =
Debug|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.Debug.Build.0 = Debug|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.IceCAP.ActiveCfg =
Release|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.IceCAP.Build.0 =
Release|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.Release.ActiveCfg =
Release|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.Release.Build.0 =
Release|Win32
    {FA0E8F97-BCE7-4DDD-A6AE-DD53F31DC960}.Release.Build.0 =
    EndGlobalSection
    GlobalSection(ExtensibilityGlobals) = postSolution
    EndGlobalSection
    GlobalSection(ExtensibilityAddIns) = postSolution
    EndGlobalSection
EndGlobal

```

Stored Procedures

delivery.sql

```

--
-- File: DELIVERY.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates delivery stored procedure
--
-- Interface Level: 4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_delivery' )
    DROP PROCEDURE tpcc_delivery
GO

CREATE PROC tpcc_delivery
    @w_id int,
    @o_carrier_id smallint

AS
DECLARE @d_id tinyint,
    @o_id int,
    @c_id int,
    @total money,
    @oid1 int,
    @oid2 int,
    @oid3 int,
    @oid4 int,
    @oid5 int,
    @oid6 int,
    @oid7 int,
    @oid8 int,
    @oid9 int,
    @oid10 int

SELECT @d_id = 0

BEGIN TRANSACTION d
    WHILE (@d_id < 10)
    BEGIN
        SELECT @d_id = @d_id + 1,
            @total = 0,
            @o_id = 0

        SELECT TOP 1

```

```

    @o_id = no_o_id
FROM   new_order WITH (serializable uplock)
WHERE  no_w_id = @w_id AND
       no_d_id = @d_id
ORDER BY no_o_id ASC

```

```

IF (@@rowcount <> 0)
BEGIN

```

```

    -- claim the order for this district
    DELETE new_order
    WHERE  no_w_id = @w_id AND
           no_d_id = @d_id AND
           no_o_id = @o_id

```

```

    -- set carrier_id on this order (and get customer id)
    UPDATE orders
    SET   o_carrier_id = @o_carrier_id,
         @c_id        = o_c_id
    WHERE o_w_id      = @w_id AND
         o_d_id      = @d_id AND
         o_id        = @o_id

```

```

    -- set date in all lineitems for this order (and sum amounts)
    UPDATE order_line
    SET   ol_delivery_d = GETDATE(),
         @total        = @total + ol_amount
    WHERE ol_w_id      = @w_id AND
         ol_d_id      = @d_id AND
         ol_o_id      = @o_id

```

```

    -- accumulate lineitem amounts for this order into customer
    UPDATE customer
    SET   c_balance    = c_balance + @total,
         c_delivery_cnt = c_delivery_cnt + 1
    WHERE c_w_id      = @w_id AND
         c_d_id      = @d_id AND
         c_id        = @c_id

```

```

END

```

```

SELECT @oid1 = CASE @d_id WHEN 1 THEN @o_id ELSE @oid1 END,
       @oid2 = CASE @d_id WHEN 2 THEN @o_id ELSE @oid2 END,
       @oid3 = CASE @d_id WHEN 3 THEN @o_id ELSE @oid3 END,
       @oid4 = CASE @d_id WHEN 4 THEN @o_id ELSE @oid4 END,
       @oid5 = CASE @d_id WHEN 5 THEN @o_id ELSE @oid5 END,
       @oid6 = CASE @d_id WHEN 6 THEN @o_id ELSE @oid6 END,
       @oid7 = CASE @d_id WHEN 7 THEN @o_id ELSE @oid7 END,
       @oid8 = CASE @d_id WHEN 8 THEN @o_id ELSE @oid8 END,
       @oid9 = CASE @d_id WHEN 9 THEN @o_id ELSE @oid9 END,
       @oid10 = CASE @d_id WHEN 10 THEN @o_id ELSE @oid10 END

```

```

END

```

```

COMMIT TRANSACTION d

```

```
-- return delivery data to client
```

```
SELECT @oid1,  
       @oid2,  
       @oid3,  
       @oid4,  
       @oid5,  
       @oid6,  
       @oid7,  
       @oid8,  
       @oid9,  
       @oid10
```

```
GO
```

```
SET QUOTED_IDENTIFIER OFF  
GO
```

```
SET ANSI_NULLS ON  
GO
```

NewOrd.sql

```
-----  
--                                     --  
-- File: NEWORD.SQL                   --  
--   Microsoft TPC-C Benchmark Kit Ver. 4.63   --  
--   Copyright Microsoft, 2005               --  
--                                     --  
--   Creates neworder stored procedure       --  
--                                     --  
--   Interface Level:  4.20.000           --  
--                                     --  
-----
```

```
SET QUOTED_IDENTIFIER OFF  
GO
```

```
SET ANSI_NULLS ON  
GO
```

```
USE tpcc  
GO
```

```
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder' )  
  DROP PROCEDURE tpcc_neworder  
GO
```

```
CREATE PROCEDURE tpcc_neworder  
    @w_id    int,  
    @d_id    tinyint,  
    @c_id    int,  
    @o_ol_cnt tinyint,  
    @o_all_local tinyint,
```



```

    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

```

AS

```

DECLARE @w_tax      smallmoney,
        @d_tax      smallmoney,
        @c_last     char(16),
        @c_credit   char(2),
        @c_discount smallmoney,
        @i_price    smallmoney,
        @i_name     char(24),
        @i_data     char(50),
        @o_entry_d  datetime,
        @remote_flag int,
        @s_quantity smallint,
        @s_data     char(50),
        @s_dist     char(24),
        @li_no      int,
        @o_id       int,
        @commit_flag tinyint,
        @li_id      int,
        @li_s_w_id  int,
        @li_qty     smallint,
        @ol_number  int,
        @c_id_local int

```

BEGIN

BEGIN TRANSACTION n

```

-----
-- get district tax and next available order id and update
-- plus initialize local variables
-----

```

```

UPDATE district
SET  @d_tax      = d_tax,
     @o_id       = d_next_o_id,
     d_next_o_id = d_next_o_id + 1,
     @o_entry_d  = GETDATE(),
     @li_no      = 0,

```

```

        @commit_flag = 1
WHERE   d_w_id      = @w_id AND
        d_id        = @d_id

-----
-- process orderlines
-----
WHILE (@li_no < @o_ol_cnt)
BEGIN
    SELECT @li_no = @li_no + 1

-----
-- set i_id, s_w_id, and qty for this lineitem
-----
    SELECT @li_id = CASE @li_no
        WHEN 1 THEN @i_id1
        WHEN 2 THEN @i_id2
        WHEN 3 THEN @i_id3
        WHEN 4 THEN @i_id4
        WHEN 5 THEN @i_id5
        WHEN 6 THEN @i_id6
        WHEN 7 THEN @i_id7
        WHEN 8 THEN @i_id8
        WHEN 9 THEN @i_id9
        WHEN 10 THEN @i_id10
        WHEN 11 THEN @i_id11
        WHEN 12 THEN @i_id12
        WHEN 13 THEN @i_id13
        WHEN 14 THEN @i_id14
        WHEN 15 THEN @i_id15
        END,

    @li_s_w_id = CASE @li_no
        WHEN 1 THEN @s_w_id1
        WHEN 2 THEN @s_w_id2
        WHEN 3 THEN @s_w_id3
        WHEN 4 THEN @s_w_id4
        WHEN 5 THEN @s_w_id5
        WHEN 6 THEN @s_w_id6
        WHEN 7 THEN @s_w_id7
        WHEN 8 THEN @s_w_id8
        WHEN 9 THEN @s_w_id9
        WHEN 10 THEN @s_w_id10
        WHEN 11 THEN @s_w_id11
        WHEN 12 THEN @s_w_id12
        WHEN 13 THEN @s_w_id13
        WHEN 14 THEN @s_w_id14
        WHEN 15 THEN @s_w_id15
        END,

    @li_qty = CASE @li_no
        WHEN 1 THEN @ol_qty1
        WHEN 2 THEN @ol_qty2

```

```

        WHEN 3 THEN @ol_qty3
        WHEN 4 THEN @ol_qty4
        WHEN 5 THEN @ol_qty5
        WHEN 6 THEN @ol_qty6
        WHEN 7 THEN @ol_qty7
        WHEN 8 THEN @ol_qty8
        WHEN 9 THEN @ol_qty9
        WHEN 10 THEN @ol_qty10
        WHEN 11 THEN @ol_qty11
        WHEN 12 THEN @ol_qty12
        WHEN 13 THEN @ol_qty13
        WHEN 14 THEN @ol_qty14
        WHEN 15 THEN @ol_qty15
    END

```

```

-----
-- get item data (no one updates item)
-----

```

```

    SELECT @i_price = i_price,
           @i_name   = i_name,
           @i_data   = i_data
    FROM   item WITH (repeatableread)
    WHERE  i_id      = @li_id

```

```

-----
-- update stock values
-----

```

```

    UPDATE stock
    SET   s_ytd      = s_ytd + @li_qty,
         @s_quantity = s_quantity - @li_qty +
             CASE WHEN (s_quantity - @li_qty < 10) THEN 91 ELSE 0 END,
         s_order_cnt = s_order_cnt + 1,
         s_remote_cnt = s_remote_cnt +
             CASE WHEN (@li_s_w_id = @w_id) THEN 0 ELSE 1 END,
         @s_data     = s_data,
         @s_dist     = CASE @d_id
             WHEN 1 THEN s_dist_01
             WHEN 2 THEN s_dist_02
             WHEN 3 THEN s_dist_03
             WHEN 4 THEN s_dist_04
             WHEN 5 THEN s_dist_05
             WHEN 6 THEN s_dist_06
             WHEN 7 THEN s_dist_07
             WHEN 8 THEN s_dist_08
             WHEN 9 THEN s_dist_09
             WHEN 10 THEN s_dist_10
         END
    WHERE s_i_id     = @li_id AND
         s_w_id     = @li_s_w_id

```

```

-----
-- if there actually is a stock (and item) with these ids, go to work
-----

```

```

IF (@@rowcount > 0)
BEGIN
-----
-- insert order_line data (using data from item and stock)
-----
INSERT INTO order_line VALUES( @o_id,
                                @d_id,
                                @w_id,
                                @li_no,
                                @li_id,
                                'dec 31, 1899',
                                @i_price * @li_qty,
                                @li_s_w_id,
                                @li_qty,
                                @s_dist)

-----
-- send line-item data to client
-----
SELECT @i_name,
       @s_quantity,
       b_g = CASE WHEN ( (patindex('%ORIGINAL%',@i_data) > 0) AND
                        (patindex('%ORIGINAL%',@s_data) >
0) )
                                THEN 'B' ELSE 'G' END,
       @i_price,
       @i_price * @li_qty
       END
       ELSE
       BEGIN
-----
-- no item (or stock) found - triggers rollback condition
-----
SELECT ",0",0,0
SELECT @commit_flag = 0
       END
       END

-----
-- get customer last name, discount, and credit rating
-----
SELECT @c_last = c_last,
       @c_discount = c_discount,
       @c_credit = c_credit,
       @c_id_local = c_id
FROM customer WITH (repeatableread)
WHERE c_id = @c_id AND
      c_w_id = @w_id AND
      c_d_id = @d_id

-----
-- insert fresh row into orders table
-----
INSERT INTO orders VALUES ( @o_id,

```

```

        @d_id,
        @w_id,
        @c_id_local,
        0,
        @o_ol_cnt,
        @o_all_local,
        @o_entry_d)

-----
-- insert corresponding row into new-order table
-----
INSERT INTO new_order VALUES ( @o_id,
                               @d_id,
                               @w_id)

-----
-- select warehouse tax
-----
SELECT @w_tax = w_tax
FROM   warehouse WITH (repeatableread)
WHERE  w_id = @w_id

IF (@commit_flag = 1)
    COMMIT TRANSACTION n
ELSE
-----
-- all that work for nuthin!!!
-----
ROLLBACK TRANSACTION n

-----
-- return order data to client
-----
SELECT @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag
END
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

null-txns.sql

```
-----
--
-- File:  NULL-TXNS.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- This script will create stored procs which
-- accept the same parameters and return correctly
-- formed results sets to match the standard TPC-C
-- stored procs. Of course, the advantage is that
-- these stored procs place almost no load on
-- SQL Server and do not require a database.
--
-- Interface Level:  4.10.000
--
-----
USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_delivery' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_orderstatus' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_payment' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_stocklevel' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_version' )
    DROP PROCEDURE tpcc_neworder
GO
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'order_line_null' )
    DROP PROCEDURE order_line_null
GO

CREATE PROCEDURE tpcc_delivery
    @w_id int,
    @o_carrier_id smallint

AS
DECLARE @d_id tinyint,
    @o_id int,
    @c_id int,
    @total numeric(12,2),
    @oid1 int,
    @oid2 int,
```

```

@oid3    int,
@oid4    int,
@oid5    int,
@oid6    int,
@oid7    int,
@oid8    int,
@oid9    int,
@oid10   int,
@delaytime varchar(30)

-----
-- uniform random delay of 0 - 1 second; avg = 0.50
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*1.00) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001, 3001
GO

CREATE PROCEDURE tpcc_neworder
    @w_id    int,
    @d_id    tinyint,
    @c_id    int,
    @o_ol_cnt tinyint,
    @o_all_local tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

AS
DECLARE @w_tax    numeric(4,4),
    @d_tax    numeric(4,4),
    @c_last    char(16),
    @c_credit    char(2),
    @c_discount    numeric(4,4),
    @i_price    numeric(5,2),
    @i_name    char(24),
    @o_entry_d    datetime,
    @li_no    int,
    @o_id    int,

```

```

    @commit_flag tinyint,
    @li_id        int,
    @li_qty       smallint,
    @delaytime    varchar(30)

BEGIN
-----
-- uniform random delay of 0 - 0.6 second; avg = 0.3
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.60) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

-----
-- process orderlines
-----
SELECT @commit_flag = 1,
       @li_no        = 0

WHILE (@li_no < @o_ol_cnt)
BEGIN
    SELECT @li_id = CASE @li_no
        WHEN 1 THEN @i_id1
        WHEN 2 THEN @i_id2
        WHEN 3 THEN @i_id3
        WHEN 4 THEN @i_id4
        WHEN 5 THEN @i_id5
        WHEN 6 THEN @i_id6
        WHEN 7 THEN @i_id7
        WHEN 8 THEN @i_id8
        WHEN 9 THEN @i_id9
        WHEN 10 THEN @i_id10
        WHEN 11 THEN @i_id11
        WHEN 12 THEN @i_id12
        WHEN 13 THEN @i_id13
        WHEN 14 THEN @i_id14
        WHEN 15 THEN @i_id15
    END

    SELECT @li_no = @li_no + 1

    SELECT @i_price = 23.45, @li_qty = @li_no

    IF (@li_id = 999999)
    BEGIN
        SELECT ",0",0,0

        SELECT @commit_flag = 0
    END
    ELSE
    BEGIN
        SELECT 'Item Name blah',
              17,

```



```

        'G',
        @i_price,
        @i_price * @li_qty
    END
END

-----
-- return order data to client
-----
SELECT @w_tax    = 0.1234,
       @d_tax    = 0.0987,
       @o_id     = 3001,
       @c_last   = 'BAROUGHTABLE',
       @c_discount = 0.2198,
       @c_credit = 'GC',
       @o_entry_d = GETDATE()

SELECT @w_tax,
       @d_tax,
       @o_id,
       @c_last,
       @c_discount,
       @c_credit,
       @o_entry_d,
       @commit_flag

END
GO

CREATE PROCEDURE tpcc_orderstatus
    @w_id    int,
    @d_id    tinyint,
    @c_id    int,
    @c_last  char(16) = ""

AS
DECLARE @c_balance    numeric(12,2),
        @c_first     char(16),
        @c_middle    char(2),
        @o_id        int,
        @o_entry_d   datetime,
        @o_carrier_id smallint,
        @ol_cnt      smallint,
        @delaytime   varchar(30)

-----
-- uniform random delay of 0 - 0.2 second; avg = 0.1
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT @c_id    = 113,

```

```

@c_balance = -10.00,
@c_first  = '8YCodgytqCj8',
@c_middle = 'OE',
@c_last   = 'OUGHTOUGHTABLE',
@o_id     = 3456,
@o_entry_d = GETDATE(),
@o_carrier_id = 1

```

```
SELECT @ol_cnt = (RAND() * 11) + 5
```

```
SET ROWCOUNT @ol_cnt
```

```

SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM   order_line_null

```

```

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,
       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id

```

```
GO
```

```

CREATE PROCEDURE tpcc_payment
    @w_id    int,
    @c_w_id  int,
    @h_amount numeric(6,2),
    @d_id    tinyint,
    @c_d_id  tinyint,
    @c_id    int,
    @c_last  char(16) = "

```

```
AS
```

```

DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city     char(20),
        @w_state    char(2),
        @w_zip      char(9),
        @w_name     char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city     char(20),
        @d_state    char(2),
        @d_zip      char(9),
        @d_name     char(10),
        @c_first    char(16),
        @c_middle   char(2),

```

```

@c_street_1 char(20),
@c_street_2 char(20),
@c_city char(20),
@c_state char(2),
@c_zip char(9),
@c_phone char(16),
@c_since datetime,
@c_credit char(2),
@c_credit_lim numeric(12,2),
@c_balance numeric(12,2),
@c_discount numeric(4,4),
@data char(500),
@c_data char(500),
@datetime datetime,
@w_ytd numeric(12,2),
@d_ytd numeric(12,2),
@cnt smallint,
@val smallint,
@screen_data char(200),
@d_id_local tinyint,
@w_id_local int,
@c_id_local int,
@delaytime varchar(30)

```

```

-----
-- uniform random delay of 0 - 0.3 second; avg = 0.15
-----

```

```

SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

```

```

WAITFOR delay @delaytime

```

```

SELECT @screen_data = "

```

```

-----
-- get customer info and update balances
-----

```

```

SELECT @d_street_1 = 'rqSHHakqyV',
       @d_street_2 = 'zZ98nW3BR2s',
       @d_city = 'ArNr4GNFV9',
       @d_state = 'aV',
       @d_zip = '453511111'

```

```

-----
-- get warehouse data and update year-to-date
-----

```

```

SELECT @w_street_1 = 'rqSHHakqyV',
       @w_street_2 = 'zZ98nW3BR2s',
       @w_city = 'ArNr4GNFV9',
       @w_state = 'aV',
       @w_zip = '453511111'

```

```

SELECT @c_id = 123,
       @c_balance = -10000.00,

```

```

@c_first = 'KmR03Xureb',
@c_middle = 'OE',
@c_last = 'BAROUGHTBAR',
@c_street_1 = 'QpGdOHjv8mR9vNI8V',
@c_street_2 = 'dzKoCObBqbC3yu',
@c_city = 'zAKZXdC037FQxq',
@c_state = 'QA',
@c_zip = '700311111',
@c_phone = '2967264064528555',
@c_credit = 'GC',
@c_credit_lim = 50000.00,
@c_discount = 0.3069,
@c_since = GETDATE(),
@datetime = GETDATE()

```

```
-----
-- return data to client
-----
```

```

SELECT @c_id,
       @c_last,
       @datetime,
       @w_street_1,
       @w_street_2,
       @w_city,
       @w_state,
       @w_zip,
       @d_street_1,
       @d_street_2,
       @d_city,
       @d_state,
       @d_zip,
       @c_first,
       @c_middle,
       @c_street_1,
       @c_street_2,
       @c_city,
       @c_state,
       @c_zip,
       @c_phone,
       @c_since,
       @c_credit,
       @c_credit_lim,
       @c_discount,
       @c_balance,
       @screen_data

```

```
GO
```

```

CREATE PROCEDURE tpcc_stocklevel
    @w_id int,
    @d_id tinyint,
    @threshold smallint

```

```
AS
```

```
DECLARE @delaytime varchar(30)
```

```

-----
-- uniform random delay of 0 - 3.6 second; avg = 1.8
-----
SELECT @delaytime = '00:00:0' + CAST(CAST((RAND()*0.20) AS decimal(4,3)) AS char(5))

WAITFOR delay @delaytime

SELECT 49
GO

CREATE PROCEDURE tpcc_version

AS
DECLARE @version char(8)

BEGIN
    SELECT @version = '4.10.000'

    SELECT @version AS 'Version'
END
GO

CREATE TABLE order_line_null (
    [ol_i_id] [int] NOT NULL ,
    [ol_supply_w_id] [int] NOT NULL ,
    [ol_delivery_d] [datetime] NOT NULL ,
    [ol_quantity] [smallint] NOT NULL ,
    [ol_amount] [numeric](6, 2) NOT NULL
) ON [PRIMARY]
GO

INSERT INTO order_line_null VALUES ( 101, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 102, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 103, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 104, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 105, 1, GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 106, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 107, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 108, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 109, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 110, 1, GETDATE(), 5, 123.45 )
INSERT INTO order_line_null VALUES ( 111, 1, GETDATE(), 1, 123.45 )
INSERT INTO order_line_null VALUES ( 112, 1, GETDATE(), 2, 123.45 )
INSERT INTO order_line_null VALUES ( 113, 1, GETDATE(), 3, 123.45 )
INSERT INTO order_line_null VALUES ( 114, 1, GETDATE(), 4, 123.45 )
INSERT INTO order_line_null VALUES ( 115, 1, GETDATE(), 5, 123.45 )
GO

```

ordstat.sql

```

--
-- File: ORDSTAT.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates order status stored procedure
--
-- Interface Level: 4.20.000
--
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_orderstatus' )
    DROP PROCEDURE tpcc_orderstatus
GO

CREATE PROCEDURE tpcc_orderstatus
    @w_id int,
    @d_id tinyint,
    @c_id int,
    @c_last char(16) = ""

AS
DECLARE @c_balance money,
    @c_first char(16),
    @c_middle char(2),
    @o_id int,
    @o_entry_d datetime,
    @o_carrier_id smallint,
    @cnt smallint

BEGIN TRANSACTION o
IF (@c_id = 0)
BEGIN
-----
-- get customer id and info using last name
-----
SELECT @cnt = (count(*)+1)/2
FROM customer WITH (repeatableread)
WHERE c_last = @c_last AND
    c_w_id = @w_id AND
    c_d_id = @d_id

SET rowcount @cnt

SELECT @c_id = c_id,

```

```

        @c_balance = c_balance,
        @c_first   = c_first,
        @c_last    = c_last,
        @c_middle  = c_middle
FROM   customer WITH (repeatableread)
WHERE  c_last     = @c_last AND
       c_w_id     = @w_id AND
       c_d_id     = @d_id
ORDER BY c_w_id, c_d_id, c_last, c_first

```

```

SET rowcount 0

```

```

END

```

```

ELSE

```

```

BEGIN

```

```

-----
-- get customer info if by id
-----

```

```

SELECT @c_balance = c_balance,
       @c_first   = c_first,
       @c_middle  = c_middle,
       @c_last    = c_last
FROM   customer WITH (repeatableread)
WHERE  c_id       = @c_id AND
       c_d_id     = @d_id AND
       c_w_id     = @w_id

```

```

SELECT @cnt       = @@rowcount
END

```

```

-----
-- if no such customer
-----

```

```

IF (@cnt = 0)

```

```

BEGIN

```

```

    RAISERROR('Customer not found',18,1)
    GOTO custnotfound

```

```

END

```

```

-----
-- get order info
-----

```

```

SELECT @o_id       = o_id,
       @o_entry_d  = o_entry_d,
       @o_carrier_id = o_carrier_id
FROM   orders WITH (serializable)
WHERE  o_c_id      = @c_id AND
       o_d_id      = @d_id AND
       o_w_id      = @w_id
ORDER BY o_id ASC

```

```

-----
-- select order lines for the current order
-----

```

```

SELECT ol_supply_w_id,
       ol_i_id,
       ol_quantity,
       ol_amount,
       ol_delivery_d
FROM   order_line WITH (repeatableread)
WHERE  ol_o_id = @o_id AND
       ol_d_id = @d_id AND
       ol_w_id = @w_id

```

custnotfound:

```

COMMIT TRANSACTION o

```

```

-----
-- return data to client
-----

```

```

SELECT @c_id,
       @c_last,
       @c_first,
       @c_middle,
       @o_entry_d,
       @o_carrier_id,
       @c_balance,
       @o_id

```

GO

payment.sql

```

-----
--                                     --
-- File: PAYMENT.SQL                   --
--   Microsoft TPC-C Benchmark Kit Ver. 4.63   --
--   Copyright Microsoft, 2005             --
--                                     --
--   Creates payment stored procedure       --
--                                     --
--   Interface Level:  4.20.000           --
--                                     --
-----

```

```

SET QUOTED_IDENTIFIER OFF
GO

```

```

SET ANSI_NULLS ON
GO

```

```

USE tpcc
GO

```

```

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_payment' )
  DROP PROCEDURE tpcc_payment
GO

```



```

CREATE PROCEDURE tpc_payment
    @w_id int,
    @c_w_id int,
    @h_amount smallmoney,
    @d_id tinyint,
    @c_d_id tinyint,
    @c_id int,
    @c_last char(16) = ""

```

AS

```

DECLARE @w_street_1 char(20),
        @w_street_2 char(20),
        @w_city char(20),
        @w_state char(2),
        @w_zip char(9),
        @w_name char(10),
        @d_street_1 char(20),
        @d_street_2 char(20),
        @d_city char(20),
        @d_state char(2),
        @d_zip char(9),
        @d_name char(10),
        @c_first char(16),
        @c_middle char(2),
        @c_street_1 char(20),
        @c_street_2 char(20),
        @c_city char(20),
        @c_state char(2),
        @c_zip char(9),
        @c_phone char(16),
        @c_since datetime,
        @c_credit char(2),
        @c_credit_lim money,
        @c_balance money,
        @c_discount smallmoney,
        @c_data char(42),
        @datetime datetime,
        @w_ytd money,
        @d_ytd money,
        @cnt smallint,
        @val smallint,
        @screen_data char(200),
        @d_id_local tinyint,
        @w_id_local int,
        @c_id_local int

```

```

SELECT @screen_data = ""

```

```

BEGIN TRANSACTION p
-- get payment date
SELECT @datetime = GETDATE()

```

```

IF (@c_id = 0)
BEGIN
    -- get customer id and info using last name
    SELECT @cnt = COUNT(*)
    FROM customer WITH (repeatableread)
    WHERE c_last = @c_last AND
           c_w_id = @c_w_id AND
           c_d_id = @c_d_id

    SELECT @val = (@cnt + 1) / 2

    SET rowcount @val

    SELECT @c_id = c_id
    FROM customer WITH (repeatableread)
    WHERE c_last = @c_last AND
           c_w_id = @c_w_id AND
           c_d_id = @c_d_id

    ORDER BY c_last, c_first

    SET rowcount 0
END

-- get customer info and update balances
UPDATE customer
SET @c_balance = c_balance = c_balance - @h_amount,
    c_payment_cnt = c_payment_cnt + 1,
    c_ytd_payment = c_ytd_payment + @h_amount,
    @c_first = c_first,
    @c_middle = c_middle,
    @c_last = c_last,
    @c_street_1 = c_street_1,
    @c_street_2 = c_street_2,
    @c_city = c_city,
    @c_state = c_state,
    @c_zip = c_zip,
    @c_phone = c_phone,
    @c_credit = c_credit,
    @c_credit_lim = c_credit_lim,
    @c_discount = c_discount,
    @c_since = c_since,
    @c_id_local = c_id
WHERE c_id = @c_id AND
       c_w_id = @c_w_id AND
       c_d_id = @c_d_id

-- if customer has bad credit get some more info
IF (@c_credit = "BC")
BEGIN
    -- compute new info
    SELECT @c_data = convert(char(5),@c_id) +
                  convert(char(4),@c_d_id) +
                  convert(char(5),@c_w_id) +

```

```

        convert(char(4),@d_id) +
        convert(char(5),@w_id) +
        convert(char(19),@h_amount)

-- update customer info
UPDATE customer
SET   c_data      = @c_data + substring(c_data, 1, 458),
      @screen_data = @c_data + substring(c_data, 1, 158)
WHERE c_id       = @c_id AND
      c_w_id     = @c_w_id AND
      c_d_id     = @c_d_id
END

-- get district data and update year-to-date
UPDATE district
SET   d_ytd      = d_ytd + @h_amount,
      @d_street_1 = d_street_1,
      @d_street_2 = d_street_2,
      @d_city     = d_city,
      @d_state    = d_state,
      @d_zip      = d_zip,
      @d_name     = d_name,
      @d_id_local = d_id
WHERE d_w_id     = @w_id AND
      d_id       = @d_id

-- get warehouse data and update year-to-date
UPDATE warehouse
SET   w_ytd      = w_ytd + @h_amount,
      @w_street_1 = w_street_1,
      @w_street_2 = w_street_2,
      @w_city     = w_city,
      @w_state    = w_state,
      @w_zip      = w_zip,
      @w_name     = w_name,
      @w_id_local = w_id
WHERE w_id       = @w_id

-- create history record
INSERT INTO history VALUES (@c_id_local,
                           @c_d_id,
                           @c_w_id,
                           @d_id_local,
                           @w_id_local,
                           @datetime,
                           @h_amount,
                           @w_name + ' ' + @d_name)

COMMIT TRANSACTION p

-- return data to client
SELECT @c_id,
       @c_last,

```

```

    @datetime,
    @w_street_1,
    @w_street_2,
    @w_city,
    @w_state,
    @w_zip,
    @d_street_1,
    @d_street_2,
    @d_city,
    @d_state,
    @d_zip,
    @c_first,
    @c_middle,
    @c_street_1,
    @c_street_2,
    @c_city,
    @c_state,
    @c_zip,
    @c_phone,
    @c_since,
    @c_credit,
    @c_credit_lim,
    @c_discount,
    @c_balance,
    @screen_data
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

StockLev.sql

```

-----
--                                     --
-- File: STOCKLEV.SQL                 --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --
-- Copyright Microsoft, 2005         --
--                                     --
-- Creates stock level stored procedure --
--                                     --
-- Interface Level: 4.20.000          --
--                                     --
-----
SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

```

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_stocklevel' )
    DROP PROCEDURE tpcc_stocklevel
GO

CREATE PROCEDURE tpcc_stocklevel
    @w_id      int,
    @d_id      tinyint,
    @threshold  smallint

AS
DECLARE @o_id_low int,
        @o_id_high int

SELECT @o_id_low = (d_next_o_id - 20),
       @o_id_high = (d_next_o_id - 1)
FROM district
WHERE d_w_id = @w_id AND
      d_id = @d_id

SELECT COUNT(DISTINCT(s_i_id))
FROM stock,
      order_line
WHERE ol_w_id = @w_id AND
      ol_d_id = @d_id and
      ol_o_id BETWEEN @o_id_low AND
                  @o_id_high AND
      s_w_id = ol_w_id AND
      s_i_id = ol_i_id AND
      s_quantity < @threshold
OPTION(ORDER GROUP)
GO

SET QUOTED_IDENTIFIER OFF
GO

SET ANSI_NULLS ON
GO

```

tpcc_neworder_new.sql

```

-----
--
-- File: TPCC_NEWORDER_NEW.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- This acid stored procedure implements the neworder
-- transaction. It outputs timestamps at the
-- beginning of the transaction, before the commit
--

```

```

--      delay, and after the commit.      --
--
-----
SET QUOTED_IDENTIFIER OFF
GO
SET ANSI_NULLS OFF
GO

USE tpcc
GO

IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_neworder_new' )
  DROP PROCEDURE tpcc_neworder_new
GO

-- neworder_new v2.5 6/23/05 PeterCa
-- lq stock/order_line/client. upd district & ins neworder.
-- cust/warehouse select together, ins order separate
-- uses rownumber to distinct w any transform
-- uses in-memory sort for distinct on iid,wid
-- uses charindex
-- will rollback if (@i_idX,@s_w_idX pairs not unique) OR (@i_idX not unique).

CREATE PROCEDURE  tpcc_neworder_new
    @w_id      int,
    @d_id      tinyint,
    @c_id      int,
    @o_ol_cnt  tinyint,
    @o_all_local  tinyint,
    @i_id1 int = 0, @s_w_id1 int = 0, @ol_qty1 smallint = 0,
    @i_id2 int = 0, @s_w_id2 int = 0, @ol_qty2 smallint = 0,
    @i_id3 int = 0, @s_w_id3 int = 0, @ol_qty3 smallint = 0,
    @i_id4 int = 0, @s_w_id4 int = 0, @ol_qty4 smallint = 0,
    @i_id5 int = 0, @s_w_id5 int = 0, @ol_qty5 smallint = 0,
    @i_id6 int = 0, @s_w_id6 int = 0, @ol_qty6 smallint = 0,
    @i_id7 int = 0, @s_w_id7 int = 0, @ol_qty7 smallint = 0,
    @i_id8 int = 0, @s_w_id8 int = 0, @ol_qty8 smallint = 0,
    @i_id9 int = 0, @s_w_id9 int = 0, @ol_qty9 smallint = 0,
    @i_id10 int = 0, @s_w_id10 int = 0, @ol_qty10 smallint = 0,
    @i_id11 int = 0, @s_w_id11 int = 0, @ol_qty11 smallint = 0,
    @i_id12 int = 0, @s_w_id12 int = 0, @ol_qty12 smallint = 0,
    @i_id13 int = 0, @s_w_id13 int = 0, @ol_qty13 smallint = 0,
    @i_id14 int = 0, @s_w_id14 int = 0, @ol_qty14 smallint = 0,
    @i_id15 int = 0, @s_w_id15 int = 0, @ol_qty15 smallint = 0

AS
BEGIN
DECLARE @o_id      int,
        @d_tax      smallmoney,
        @o_entry_d  datetime,
        @commit_flag  tinyint

BEGIN TRANSACTION n

```

```
-- get district tax and next available order id and update
-- insert corresponding row into new-order table
-- plus initialize local variables
```

```
UPDATE district
SET   @d_tax      = d_tax,
      @o_id       = d_next_o_id,
      d_next_o_id = d_next_o_id + 1,
      @o_entry_d  = GETDATE(),
      @commit_flag = 1
OUTPUT deleted.d_next_o_id,
       @d_id,
       @w_id
INTO   new_order
WHERE  d_w_id     = @w_id AND
       d_id      = @d_id
```

```
-- update stock from stock join (item join (params))
-- output to orderline, output to client
-- NOTE: @@rowcount != @ol_o_cnt
-- if (@i_idX,@s_w_idX pairs not unique) OR (@i_idX not unique).
```

```
UPDATE stock
SET   s_ytd      = s_ytd + info.ol_qty,
      s_quantity = s_quantity - info.ol_qty +
                CASE WHEN (s_quantity - info.ol_qty < 10) THEN 91 ELSE 0 END,
      s_order_cnt = s_order_cnt + 1,
      s_remote_cnt = s_remote_cnt +
                CASE WHEN (info.w_id = @w_id) THEN 0 ELSE 1 END
```

```
OUTPUT @o_id,
       @d_id,
       @w_id,
       info.lino,
       info.i_id,
       "dec 31, 1899",
       info.i_price * info.ol_qty,
       info.w_id,
       info.ol_qty,
       @d_id WHEN 1 THEN inserted.s_dist_01
             WHEN 2 THEN inserted.s_dist_02
             WHEN 3 THEN inserted.s_dist_03
             WHEN 4 THEN inserted.s_dist_04
             WHEN 5 THEN inserted.s_dist_05
             WHEN 6 THEN inserted.s_dist_06
             WHEN 7 THEN inserted.s_dist_07
             WHEN 8 THEN inserted.s_dist_08
             WHEN 9 THEN inserted.s_dist_09
             WHEN 10 THEN inserted.s_dist_10
END
INTO   order_line
```

```
OUTPUT info.i_name,inserted.s_quantity,
```

```

CASE WHEN ((charindex("ORIGINAL",info.i_data) > 0) AND
(charindex("ORIGINAL",inserted.s_data) > 0) )
THEN "B" ELSE "G" END,
info.i_price,
info.i_price*info.ol_qty
FROM stock INNER JOIN
(SELECT iid,
wid,
lino,
ol_qty,
i_price,
i_name,
i_data
FROM (SELECT iid,
wid,
lino,
qty,
row_number() OVER (PARTITION BY iid,wid ORDER BY iid,wid)
FROM (SELECT @i_id1,@s_w_id1,1,@ol_qty1 UNION ALL
SELECT @i_id2,@s_w_id2,2,@ol_qty2 UNION ALL
SELECT @i_id3,@s_w_id3,3,@ol_qty3 UNION ALL
SELECT @i_id4,@s_w_id4,4,@ol_qty4 UNION ALL
SELECT @i_id5,@s_w_id5,5,@ol_qty5 UNION ALL
SELECT @i_id6,@s_w_id6,6,@ol_qty6 UNION ALL
SELECT @i_id7,@s_w_id7,7,@ol_qty7 UNION ALL
SELECT @i_id8,@s_w_id8,8,@ol_qty8 UNION ALL
SELECT @i_id9,@s_w_id9,9,@ol_qty9 UNION ALL
SELECT @i_id10,@s_w_id10,10,@ol_qty10 UNION ALL
SELECT @i_id11,@s_w_id11,11,@ol_qty11 UNION ALL
SELECT @i_id12,@s_w_id12,12,@ol_qty12 UNION ALL
SELECT @i_id13,@s_w_id13,13,@ol_qty13 UNION ALL
SELECT @i_id14,@s_w_id14,14,@ol_qty14 UNION ALL
SELECT @i_id15,@s_w_id15,15,@ol_qty15) AS uo1(iid,wid,lino,qty)
) AS o1(iid,wid,lino,ol_qty,rownum)
INNER JOIN
item (repeatableread) ON i_id = iid AND -- filters out invalid items
rownum = 1
) AS info(i_id,w_id,lino,ol_qty,i_price,i_name,i_data)
ON s_i_id = info.i_id AND
s_w_id = info.w_id

```

```

IF (@@rowcount <> @o_ol_cnt) -- must have an invalid item
SELECT @commit_flag = 0 -- 2.4.2.3 requires rest to proceed

```

```

-- insert fresh row into orders table
INSERT INTO orders VALUES ( @o_id,
@d_id,
@w_id,
@c_id,
0,
@o_ol_cnt,
@o_all_local,
@o_entry_d)

```



```

-- get customer last name, discount, and credit rating
-- get warehouse tax
-- return order_data to client

SELECT w_tax,
       @d_tax,
       @o_id,
       c_last,
       c_discount,
       c_credit,
       @o_entry_d,
       @commit_flag
FROM   warehouse(repeatableread),
       customer(repeatableread)
WHERE  w_id = @w_id AND
       c_id = @c_id AND
       c_w_id = @w_id AND
       c_d_id = @d_id

-- @@rowcount checks that previous select found a valid customer
IF ((@commit_flag = 1) AND (@@rowcount = 1))
  COMMIT TRANSACTION n
ELSE -- all that work for nothing.
  ROLLBACK TRANSACTION n

END
GO

```

version.sql

```

-----
--
-- File:  VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Returns version level of TPC-C stored procs
--
-- Always update the return value of this proc for
-- any interface changes or 'must have' bug fixes.
--
-- The value returned by this SP defines the
-- 'interface level', which must match between the
-- stored procs and the client code. The
-- interface level may be down rev from the
-- current kit. This indicates that the interface
-- hasn't changed since that version.
--
-- Interface Level:  4.20.000
--

```

USE tpcc
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'tpcc_version')
 DROP PROCEDURE tpcc_version

GO

CREATE PROCEDURE tpcc_version
AS
DECLARE @version char(8)

BEGIN

SELECT @version = '4.20.000'

SELECT @version AS 'Version'

END

GO

Appendix B: DATABASE DESIGN

Database Build

backup.sql

```
-----  
--  
-- File: BACKUP.SQL --  
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --  
-- Copyright Microsoft, 2005 --  
--  
-----  
  
DECLARE @startdate DATETIME,  
        @enddate DATETIME  
  
SELECT @startdate = GETDATE()  
SELECT 'Start date:',  
       CONVERT(VARCHAR(30),@startdate, 21)  
  
DUMP DATABASE tpcc TO tpccback1, tpccback2, tpccback3, tpccback4 WITH init, stats = 1  
  
SELECT @enddate = GETDATE()  
SELECT 'End date: ',  
       CONVERT(VARCHAR(30),@enddate, 21)  
SELECT 'Elapsed time (in seconds): ',  
       DATEDIFF(second, @startdate, @enddate)  
GO
```

backupdev.sql

```
-----  
--  
-- File: BACKUPDEV.SQL --  
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --  
-- Copyright Microsoft, 2005 --  
--  
-----  
  
USE master  
GO  
  
-----  
-- create backup devices  
-----  
EXEC sp_addumpdevice 'disk','tpccback1','W:\tpccback_rtm_x64_kit463_36bays_10000wh_1.dmp'  
GO  
EXEC sp_addumpdevice 'disk','tpccback2','X:\tpccback_rtm_x64_kit463_36bays_10000wh_2.dmp'
```

```

GO
EXEC sp_addumpdevice 'disk','tpcback3','Y:\tpcback_rtm_x64_kit463_36bays_10000wh_3.dmp'
GO
EXEC sp_addumpdevice 'disk','tpcback4','Z:\tpcback_rtm_x64_kit463_36bays_10000wh_4.dmp'
GO

```

createdb.sql

```

-----
--
-- File:  CREATEDB.SQL
--      Microsoft TPC-C Benchmark Kit Ver. 4.63
--      Copyright Microsoft, 2005
--
-----

SET ANSI_NULL_DFLT_OFF ON
GO

USE master
GO

-----
-- Create temporary table for timing
-----
IF EXISTS( SELECT name FROM sysobjects WHERE name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

CREATE TABLE tpcc_timer
    (start_date    CHAR(30),
     end_date     CHAR(30))
GO

INSERT INTO tpcc_timer VALUES(0,0)
GO

-----
-- Store starting time
-----
UPDATE tpcc_timer
SET    start_date    = (SELECT CONVERT(CHAR(30), GETDATE(), 21))
GO

-----
-- create main database files
-----
CREATE DATABASE tpcc
ON PRIMARY
(
    NAME           = MSSQL_tpcc_root,
    FILENAME      = 'U:\MSSQL_tpcc_root_kit463_rtm_x64_10kwh_36bays.mdf',
    SIZE          = 8MB,

```

```

        FILEGROWTH = 0),
FILEGROUP  MSSQL_misc_fg
(
    NAME      = MSSQL_misc1,FILENAME  =      'c:\tpcc\M1\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc2,FILENAME  =      'c:\tpcc\M2\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc3,FILENAME  =      'c:\tpcc\M3\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc4,FILENAME  =      'c:\tpcc\M4\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc5,FILENAME  =      'c:\tpcc\M5\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc6,FILENAME  =      'c:\tpcc\M6\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc7,FILENAME  =      'c:\tpcc\M7\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc8,FILENAME  =      'c:\tpcc\M8\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc9,FILENAME  =      'c:\tpcc\M9\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc10,FILENAME =      'c:\tpcc\M10\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc11,FILENAME =      'c:\tpcc\M11\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc12,FILENAME =      'c:\tpcc\M12\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc13,FILENAME =      'c:\tpcc\M13\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc14,FILENAME =      'c:\tpcc\M14\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc15,FILENAME =      'c:\tpcc\M15\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc16,FILENAME =      'c:\tpcc\M16\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc17,FILENAME =      'c:\tpcc\M17\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc18,FILENAME =      'c:\tpcc\M18\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc19,FILENAME =      'c:\tpcc\M19\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc20,FILENAME =      'c:\tpcc\M20\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc21,FILENAME =      'c:\tpcc\M21\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc22,FILENAME =      'c:\tpcc\M22\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc23,FILENAME =      'c:\tpcc\M23\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc24,FILENAME =      'c:\tpcc\M24\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc38,FILENAME =      'c:\tpcc\M25\',SIZE=
10000MB,FILEGROWTH= 0),
(
    NAME      = MSSQL_misc26,FILENAME =      'c:\tpcc\M26\',SIZE=

```

```

10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc27,FILENAME = 'c:\tpcc\M27',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc28,FILENAME = 'c:\tpcc\M28',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc29,FILENAME = 'c:\tpcc\M29',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc30,FILENAME = 'c:\tpcc\M30',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc31,FILENAME = 'c:\tpcc\M31',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc32,FILENAME = 'c:\tpcc\M32',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc33,FILENAME = 'c:\tpcc\M33',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc34,FILENAME = 'c:\tpcc\M34',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc35,FILENAME = 'c:\tpcc\M35',SIZE=
10000MB,FILEGROWTH= 0),
( NAME = MSSQL_misc36,FILENAME = 'c:\tpcc\M36',SIZE=
10000MB,FILEGROWTH= 0),
FILEGROUP MSSQL_cs_fg
( NAME = MSSQL_cs1,FILENAME= 'c:\tpcc\CS1',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs2,FILENAME= 'c:\tpcc\CS2',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs3,FILENAME= 'c:\tpcc\CS3',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs4,FILENAME= 'c:\tpcc\CS4',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs5,FILENAME= 'c:\tpcc\CS5',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs6,FILENAME= 'c:\tpcc\CS6',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs7,FILENAME= 'c:\tpcc\CS7',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs8,FILENAME= 'c:\tpcc\CS8',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs9,FILENAME= 'c:\tpcc\CS9',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs10,FILENAME= 'c:\tpcc\CS10',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs11,FILENAME= 'c:\tpcc\CS11',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs12,FILENAME= 'c:\tpcc\CS12',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs13,FILENAME= 'c:\tpcc\CS13',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs14,FILENAME= 'c:\tpcc\CS14',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs15,FILENAME= 'c:\tpcc\CS15',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs16,FILENAME= 'c:\tpcc\CS16',SIZE=

```

```

18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs17,FILENAME= 'c:\tpcc\CS17\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs18,FILENAME= 'c:\tpcc\CS18\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs19,FILENAME= 'c:\tpcc\CS19\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs20,FILENAME= 'c:\tpcc\CS20\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs21,FILENAME= 'c:\tpcc\CS21\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs22,FILENAME= 'c:\tpcc\CS22\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs23,FILENAME= 'c:\tpcc\CS23\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs24,FILENAME= 'c:\tpcc\CS24\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs38,FILENAME= 'c:\tpcc\CS25\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs26,FILENAME= 'c:\tpcc\CS26\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs27,FILENAME= 'c:\tpcc\CS27\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs28,FILENAME= 'c:\tpcc\CS28\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs29,FILENAME= 'c:\tpcc\CS29\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs30,FILENAME= 'c:\tpcc\CS30\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs31,FILENAME= 'c:\tpcc\CS31\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs32,FILENAME= 'c:\tpcc\CS32\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs33,FILENAME= 'c:\tpcc\CS33\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs34,FILENAME= 'c:\tpcc\CS34\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs35,FILENAME= 'c:\tpcc\CS35\',SIZE=
18000MB,FILEGROWTH= 0),
( NAME = MSSQL_cs36,FILENAME= 'c:\tpcc\CS36\',SIZE=
18000MB,FILEGROWTH= 0)
LOG ON
( NAME = MSSQL_tpcc_log,
FILENAME = 'L:',
SIZE = 400000MB,
FILEGROWTH = 0)
COLLATE Latin1_General_BIN
GO

```

```

-----
-- Store ending time
-----

```

```

UPDATE tpcc_timer

```

```

SET end_date = (SELECT CONVERT(Char(30), GETDATE(), 21))
GO

SELECT DATEDIFF(second,(SELECT start_date FROM tpcc_timer),(SELECT end_date FROM
tpcc_timer))
GO

-----
-- remove temporary table
-----
IF EXISTS ( SELECT name FROM sysobjects WHERE name = 'tpcc_timer' )
    DROP TABLE tpcc_timer
GO

```

dropdev.sql

```

-----
--
-- File: REMOVEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-----

USE master
GO

-----

-- remove any existing backup files
-----

EXEC sp_dropdevice 'tpccback1'
EXEC sp_dropdevice 'tpccback2'
EXEC sp_dropdevice 'tpccback3'
EXEC sp_dropdevice 'tpccback4'
GO

```

removedb.sql

```

-----
--
-- File: REMOVEDB.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-----

USE master
GO

-----

-- remove any existing database and backup files

```



```
-----  
EXEC sp_dbremove tpcc, dropdev  
GO
```

```
EXEC sp_dropdevice 'tpcback1'  
EXEC sp_dropdevice 'tpcback2'  
EXEC sp_dropdevice 'tpcback3'  
EXEC sp_dropdevice 'tpcback4'  
GO
```

restore.sql

```
-----  
-- File: RESTORE.SQL --  
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --  
-- Copyright Microsoft, 2005 --  
-- --  
-----
```

```
DECLARE @startdate DATETIME,  
        @enddate DATETIME
```

```
SELECT @startdate = GETDATE()  
SELECT 'Start date:',  
       CONVERT(VARCHAR(30),@startdate, 21)
```

```
LOAD DATABASE tpcc FROM tpcback1, tpcback2, tpcback3, tpcback4 WITH replace, stats = 1
```

```
SELECT @enddate = GETDATE()  
SELECT 'End date: ',  
       CONVERT(VARCHAR(30),@enddate, 21)  
SELECT 'Elapsed time (in seconds): ',  
       DATEDIFF(second, @startdate, @enddate)  
GO
```

idxcuscl.sql

```
-----  
-- File: IDXCUSCL.SQL --  
-- Microsoft TPC-C Benchmark Kit Ver. 4.63 --  
-- Copyright Microsoft, 2005 --  
-- --  
-- Creates clustered index on customer table --  
-----
```

```
USE tpcc  
GO
```

```
DECLARE @startdate DATETIME,
```

```

        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'customer_c1' )
    DROP INDEX customer.customer_c1

CREATE UNIQUE CLUSTERED INDEX customer_c1 ON customer(c_w_id, c_d_id, c_id)
    ON MSSQL_cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxcusnc.sql

```

-----
--                                     --
-- File:  IDXCUSNC.SQL                 --
--   Microsoft TPC-C Benchmark Kit Ver. 4.63   --
--   Copyright Microsoft, 2005             --
--                                     --
--   Creates non-clustered index on customer table   --
-----
USE tpcc
GO

DECLARE @startdate  DATETIME,
        @enddate   DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'customer_nc1' )
    DROP INDEX customer.customer_nc1

CREATE UNIQUE NONCLUSTERED INDEX customer_nc1 ON customer(c_w_id, c_d_id, c_last,
c_first, c_id)
    ON MSSQL_cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxdiscl.sql

```
-----  
--  
-- File:  IDXDISCL.SQL  
--      Microsoft TPC-C Benchmark Kit Ver. 4.63  
--      Copyright Microsoft, 2005  
--  
--      Creates clustered index on district table  
-----  
USE tpcc  
GO  
  
DECLARE @startdate  DATETIME,  
        @enddate  DATETIME  
  
SELECT @startdate = GETDATE()  
SELECT 'Start date:',  
       CONVERT(VARCHAR(30),@startdate,21)  
  
IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'district_c1' )  
    DROP INDEX district.district_c1  
  
CREATE UNIQUE CLUSTERED INDEX district_c1 ON district(d_w_id, d_id)  
    WITH FILLFACTOR=100 ON MSSQL_misc_fg  
  
SELECT @enddate = GETDATE()  
SELECT 'End date:',  
       CONVERT(VARCHAR(30),@enddate,21)  
SELECT 'Elapsed time (in seconds): ',  
       DATEDIFF(second, @startdate, @enddate)  
GO
```

idxhiscl.sql

```
-----  
--  
-- File:  IDXHISCL.SQL  
--      Microsoft TPC-C Benchmark Kit Ver. 4.63  
--      Copyright Microsoft, 2005  
--  
--      Creates clustered index on history table  
--  
--      CAUTION: This index is only beneficial for systems --  
--      CAUTION: with 8 or more processors.                --  
--      CAUTION: It may negatively impact performance on  --  
--      CAUTION: systems with less than 8 processors.      --  
--  
-----
```

```

USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'history_c1' )
    DROP INDEX history.history_c1

CREATE UNIQUE CLUSTERED INDEX history_c1 ON history(h_c_w_id, h_date, h_c_d_id,
h_c_id, h_amount)
    ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxitmcl.sql

```

-----
--
-- File:  IDXITMCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates clustered index on item table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'item_c1' )
    DROP INDEX item.item_c1

CREATE UNIQUE CLUSTERED INDEX item_c1 ON item(i_id)
    ON MSSQL_misc_fg

SELECT @enddate = GETDATE()

```

```

SELECT 'End date:',
      CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
      DATEDIFF(second, @startdate, @enddate)
GO

```

idxnodcl.sql

```

-----
--
-- File:  IDXNODCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates clustered index on new-order table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
      CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'new_order_c1' )
  DROP INDEX new_order.new_order_c1

CREATE UNIQUE CLUSTERED INDEX new_order_c1 ON new_order(no_w_id, no_d_id, no_o_id)

  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
      CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
      DATEDIFF(second, @startdate, @enddate)
GO

```

idxodlcl.sql

```

-----
--
-- File:  IDXODLCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates clustered index on order-line table
--

```

```

--
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'order_line_c1' )
  DROP INDEX order_line.order_line_c1

CREATE UNIQUE CLUSTERED INDEX order_line_c1 ON order_line(ol_w_id, ol_d_id, ol_o_id,
ol_number)
  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxordcl.sql

```

-----
--
-- File:  IDXORDCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates clustered index on orders table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'orders_c1' )
  DROP INDEX orders.orders_c1

CREATE UNIQUE CLUSTERED INDEX orders_c1 ON orders(o_w_id, o_d_id, o_id)
  ON MSSQL_misc_fg

```

```

SELECT @enddate = GETDATE()
SELECT 'End date:',
      CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
      DATEDIFF(second, @startdate, @enddate)
GO

```

idxordnc.sql

```

-----
--                                     --
-- File:  IDXORDNC.SQL                 --
--   Microsoft TPC-C Benchmark Kit Ver. 4.63   --
--   Copyright Microsoft, 2005             --
--                                     --
--   Creates non-clustered index on orders table   --
-----
USE tpcc
GO

DECLARE @startdate  DATETIME,
        @enddate  DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
      CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'orders_nc1' )
  DROP INDEX orders.orders_nc1

CREATE INDEX orders_nc1 ON orders(o_w_id, o_d_id, o_c_id, o_id)
  ON MSSQL_misc_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
      CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
      DATEDIFF(second, @startdate, @enddate)
GO

```

idxstkcl.sql

```

-----
--                                     --
-- File:  IDXSTKCL.SQL                 --
--   Microsoft TPC-C Benchmark Kit Ver. 4.63   --
--   Copyright Microsoft, 2005             --
--                                     --
--   Creates clustered index on stock table   --
-----

```

```

--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'stock_c1' )
  DROP INDEX stock.stock_c1

CREATE UNIQUE CLUSTERED INDEX stock_c1 ON stock(s_i_id, s_w_id)
  ON MSSQL_cs_fg

SELECT @enddate = GETDATE()
SELECT 'End date:',
       CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
       DATEDIFF(second, @startdate, @enddate)
GO

```

idxwarcl.sql

```

-----
--
-- File: IDXWARCL.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates clustered index on warehouse table
--
-----
USE tpcc
GO

DECLARE @startdate DATETIME,
        @enddate DATETIME

SELECT @startdate = GETDATE()
SELECT 'Start date:',
       CONVERT(VARCHAR(30),@startdate,21)

IF EXISTS ( SELECT name FROM sysindexes WHERE name = 'warehouse_c1' )
  DROP INDEX warehouse.warehouse_c1

CREATE UNIQUE CLUSTERED INDEX warehouse_c1 ON warehouse(w_id)
  WITH FILLFACTOR=100 ON MSSQL_misc_fg

```



```

SELECT @enddate = GETDATE()
SELECT 'End date:',
      CONVERT(VARCHAR(30),@enddate,21)
SELECT 'Elapsed time (in seconds): ',
      DATEDIFF(second, @startdate, @enddate)
GO

```

tables.sql

```

-----
--
-- File: TABLES.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Creates TPC-C tables
-----

SET ANSI_NULL_DFLT_OFF ON
GO

USE tpcc
GO

-----
-- Remove all existing TPC-C tables
-----
if exists ( select name from sysobjects where name = 'warehouse' )
    drop table warehouse
go
if exists ( select name from sysobjects where name = 'district' )
    drop table district
go
if exists ( select name from sysobjects where name = 'customer' )
    drop table customer
go
if exists ( select name from sysobjects where name = 'history' )
    drop table history
go
if exists ( select name from sysobjects where name = 'new_order' )
    drop table new_order
go
if exists ( select name from sysobjects where name = 'orders' )
    drop table orders
go
if exists ( select name from sysobjects where name = 'order_line' )
    drop table order_line
go
if exists ( select name from sysobjects where name = 'item' )
    drop table item
go
if exists ( select name from sysobjects where name = 'stock' )

```

```

        drop table stock
go

-----
-- Create new tables
-----
create table warehouse
(
    w_id          int,
    w_ytd         money,
    w_tax         smallmoney,
    w_name        char(10),
    w_street_1    char(20),
    w_street_2    char(20),
    w_city        char(20),
    w_state       char(2),
    w_zip         char(9)
) on MSSQL_misc_fg
go

create table district
(
    d_id          tinyint,
    d_w_id        int,
    d_ytd         money,
    d_next_o_id   int,
    d_tax         smallmoney,
    d_name        char(10),
    d_street_1    char(20),
    d_street_2    char(20),
    d_city        char(20),
    d_state       char(2),
    d_zip         char(9)
) on MSSQL_misc_fg
go

create table customer
(
    c_id          int,
    c_d_id        tinyint,
    c_w_id        int,
    c_discount    smallmoney,
    c_credit_lim  money,
    c_last        char(16),
    c_first       char(16),
    c_credit      char(2),
    c_balance     money,
    c_ytd_payment money,
    c_payment_cnt smallint,
    c_delivery_cnt smallint,
    c_street_1    char(20),
    c_street_2    char(20),
    c_city        char(20),

```

```

        c_state      char(2),
        c_zip        char(9),
        c_phone      char(16),
        c_since      datetime,
        c_middle     char(2),
        c_data       char(500)
) on MSSQL_cs_fg
go

-- Use the following table option if using c_data varchar(max)
-- sp_tableoption 'customer','large value types out of row','1'
-- go

```

```

create table history
(
    h_c_id          int,
    h_c_d_id        tinyint,
    h_c_w_id        int,
    h_d_id          tinyint,
    h_w_id          int,
    h_date          datetime,
    h_amount        smallmoney,
    h_data          char(24)
) on MSSQL_misc_fg
go

```

```

create table new_order
(
    no_o_id         int,
    no_d_id         tinyint,
    no_w_id         int
) on MSSQL_misc_fg
go

```

```

create table orders
(
    o_id           int,
    o_d_id         tinyint,
    o_w_id         int,
    o_c_id         int,
    o_carrier_id   tinyint,
    o_ol_cnt       tinyint,
    o_all_local    tinyint,
    o_entry_d      datetime
) on MSSQL_misc_fg
go

```

```

create table order_line
(
    ol_o_id        int,
    ol_d_id        tinyint,
    ol_w_id        int,
    ol_number      tinyint,

```

```

        ol_i_id      int,
        ol_delivery_d  datetime,
        ol_amount     smallmoney,
        ol_supply_w_id int,
        ol_quantity   smallint,
        ol_dist_info  char(24)
    ) on MSSQL_misc_fg
go

```

```

create table item
(
    i_id      int,
    i_name    char(24),
    i_price   smallmoney,
    i_data    char(50),
    i_im_id   int
) on MSSQL_misc_fg
go

```

```

create table stock
(
    s_i_id      int,
    s_w_id      int,
    s_quantity  smallint,
    s_ytd       int,
    s_order_cnt smallint,
    s_remote_cnt smallint,
    s_data      char(50),
    s_dist_01   char(24),
    s_dist_02   char(24),
    s_dist_03   char(24),
    s_dist_04   char(24),
    s_dist_05   char(24),
    s_dist_06   char(24),
    s_dist_07   char(24),
    s_dist_08   char(24),
    s_dist_09   char(24),
    s_dist_10   char(24)
) on MSSQL_cs_fg
go

```

Load Source Code

dbopt1.sql

```

-----
--                                     --
-- File: DBOPT1.SQL                   --
-- Microsoft TPC-C Benchmark Kit Ver. 4.63   --
-- Copyright Microsoft, 2005           --

```

```

--          --
--      Sets database options for load          --
--          --
-----
USE master
GO

ALTER DATABASE tpcc SET RECOVERY BULK_LOGGED
GO

EXEC sp_dboption tpcc,'trunc. log on chkpt.',TRUE
GO

ALTER DATABASE tpcc SET TORN_PAGE_DETECTION OFF
GO

ALTER DATABASE tpcc SET PAGE_VERIFY NONE
GO

USE tpcc
GO

CHECKPOINT
GO

```

dbopt2.sql

```

-----
--          --
-- File:  DBOPT2.SQL          --
--      Microsoft TPC-C Benchmark Kit Ver. 4.63      --
--      Copyright Microsoft, 2005          --
--          --
--      Sets database options after load          --
--          --
-----
ALTER DATABASE tpcc SET RECOVERY FULL
GO

USE tpcc
GO

CHECKPOINT
GO

sp_configure 'allow updates',1
GO

RECONFIGURE WITH OVERRIDE
GO

DECLARE      @msg      varchar(50)

```

```

-----
--      OPTIONS FOR SQL SERVER 2000      --
-- Set option values for user-defined indexes --
-----

SET  @msg = ''
PRINT @msg
SET  @msg = 'Setting SQL Server indexoptions'
PRINT @msg
SET  @msg = ''
PRINT @msg

EXEC sp_indexoption 'customer', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'district', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'warehouse', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'stock', 'DisallowPageLocks', TRUE
EXEC sp_indexoption 'order_line', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'orders', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'new_order', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'item', 'DisallowRowLocks', TRUE
EXEC sp_indexoption 'item', 'DisallowPageLocks', FALSE
GO

Print ''
Print '*****'
Print 'Pre-specified Locking Hierarchy:'
Print '  Lockflag = 0 ==> No pre-specified hierarchy'
Print '  Lockflag = 1 ==> Lock at Page-level then Table-level'
Print '  Lockflag = 2 ==> Lock at Row-level then Table-level'
Print '  Lockflag = 3 ==> Lock at Table-level'
Print ''

SELECT name,
       lockflags
FROM sysindexes
WHERE object_id('warehouse') = id OR
       object_id('district') = id OR
       object_id('customer') = id OR
       object_id('stock') = id OR
       object_id('orders') = id OR
       object_id('order_line') = id OR
       object_id('history') = id OR
       object_id('new_order') = id OR
       object_id('item') = id
ORDER BY lockflags asc
GO

sp_configure 'allow updates',0
GO

RECONFIGURE WITH OVERRIDE
GO

```

```
EXEC sp_dboption tpcc, 'auto update statistics', FALSE
EXEC sp_dboption tpcc, 'auto create statistics', FALSE
GO
```

```
DECLARE @db_id int,
        @tbl_id int
```

```
SET @db_id = DB_ID('tpcc')
SET @tbl_id = OBJECT_ID('tpcc..warehouse')
DBCC PINTABLE (@db_id, @tbl_id)
```

```
SET @tbl_id = OBJECT_ID('tpcc..district')
DBCC PINTABLE (@db_id, @tbl_id)
```

```
SET @tbl_id = OBJECT_ID('tpcc..new_order')
DBCC PINTABLE (@db_id, @tbl_id)
```

```
SET @tbl_id = OBJECT_ID('tpcc..item')
DBCC PINTABLE (@db_id, @tbl_id)
GO
```

RunSQLCfg.sql

```
-----
--
-- File: RUNSQLCFG.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--
-- Sets suggested runtime server configuration
-- parameters
--
-----
EXEC sp_configure 'show advanced option', 1
GO

RECONFIGURE WITH OVERRIDE
GO

-----
-- change this value to approximately the number of connected users
-----
EXEC sp_configure 'max worker threads',255

-----
-- increase priority of user threads
-----
EXEC sp_configure 'priority boost',1

-----
-- disable automatic checkpointing
```

```
-----  
EXEC sp_configure 'recovery interval',32767
```

```
-----  
-- change to a mask appropriate for the number of processors on the server  
-----
```

```
EXEC sp_configure 'affinity mask',0xf
```

```
-----  
-- enable fibers  
-----
```

```
EXEC sp_configure 'lightweight pooling',1  
GO
```

```
RECONFIGURE WITH OVERRIDE  
GO
```

sqlshutdown.sql

```
-----  
--  
-- File:  SQLSHUTDOWN.SQL  
--      Microsoft TPC-C Benchmark Kit Ver. 4.63  
--      Copyright Microsoft, 2005  
--  
--      Checkpoints tpcc database and issues a shutdown  
--  
-----
```

```
USE tpcc  
GO
```

```
CHECKPOINT  
GO
```

```
SHUTDOWN  
GO
```

VerifyTPCCLoad.sql

```
-----  
--  
-- File:  VerifyTPCCLoad.SQL  
--      Microsoft TPC-C Benchmark Kit Ver. 4.63  
--      Copyright Microsoft, 2005  
--  
-----
```

```
SET NOCOUNT ON  
PRINT ''  
SELECT CONVERT(CHAR(30), GETDATE(), 21)  
PRINT ''
```



```

USE tpcc
GO

IF EXISTS (SELECT name
           FROM sysobjects
           WHERE name = 'TPCC_INFO' AND
                 type = 'U')
    DROP TABLE TPCC_INFO
GO
PRINT 'WAREHOUSE TABLE'
SELECT count_big(*)
FROM warehouse
GO

PRINT 'DISTRICT TABLE = (10 * No of warehouses)'
SELECT count_big(*)
FROM district
GO

PRINT 'ITEM TABLE = 100,000'
SELECT count_big(*)
FROM item
GO

PRINT 'CUSTOMER TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM customer
GO

PRINT 'ORDERS TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM orders
GO

PRINT 'HISTORY TABLE = (30,000 * No of warehouses)'
SELECT count_big(*)
FROM history
GO

PRINT 'STOCK TABLE = (100,000 * No of warehouses)'
SELECT count_big(*)
FROM stock
GO

PRINT 'ORDER_LINE TABLE = (300,000 * No of warehouses + some change)'
SELECT count_big(*)
FROM order_line
GO

PRINT 'NEW_ORDER TABLE = (9000 * No of warehouses)'
SELECT count_big(*)
FROM new_order

```

GO

CREATE TABLE TPCC_INFO

```
( INFO_DATE          datetime,
  NUM_WAREHOUSE      bigint,
  WAREHOUSE_TARGET   bigint,
  NUM_DISTRICT       bigint,
  DISTRICT_TARGET    bigint,
  NUM_ITEM            bigint,
  ITEM_TARGET        bigint,
  NUM_CUSTOMER       bigint,
  CUSTOMER_TARGET    bigint,
  NUM_ORDERS          bigint,
  ORDERS_TARGET      bigint,
  ORDERS_TARGET_LOW  bigint,
  ORDERS_TARGET_HIGH bigint,
  NUM_ORDER_LINE     bigint,
  ORDER_LINE_TARGET  bigint,
  ORDER_LINE_TARGET_LOW  bigint,
  ORDER_LINE_TARGET_HIGH  bigint,
  NUM_NEW_ORDER      bigint,
  NEW_ORDER_TARGET   bigint,
  NEW_ORDER_TARGET_LOW  bigint,
  NEW_ORDER_TARGET_HIGH  bigint,
  NUM_HISTORY        bigint,
  HISTORY_TARGET     bigint,
  NUM_STOCK          bigint,
  STOCK_TARGET       bigint)
```

GO

```
DECLARE @NUM_WAREHOUSE      bigint,
        @WAREHOUSE_TARGET   bigint,
        @NUM_DISTRICT       bigint,
        @DISTRICT_TARGET    bigint,
        @NUM_ITEM           bigint,
        @ITEM_TARGET        bigint,
        @NUM_CUSTOMER       bigint,
        @CUSTOMER_TARGET    bigint,
        @NUM_ORDERS         bigint,
        @ORDERS_TARGET      bigint,
        @ORDERS_TARGET_LOW  bigint,
        @ORDERS_TARGET_HIGH  bigint,
        @NUM_ORDER_LINE     bigint,
        @ORDER_LINE_TARGET  bigint,
        @ORDER_LINE_TARGET_LOW  bigint,
        @ORDER_LINE_TARGET_HIGH  bigint,
        @NUM_NEW_ORDER      bigint,
        @NEW_ORDER_TARGET   bigint,
        @NEW_ORDER_TARGET_LOW  bigint,
        @NEW_ORDER_TARGET_HIGH  bigint,
        @NUM_HISTORY        bigint,
        @HISTORY_TARGET     bigint,
        @NUM_STOCK          bigint,
```

@STOCK_TARGET bigint

-- set the local variables prior to inserting them into the TPCC_INFO table

```
SELECT @NUM_WAREHOUSE = COUNT_BIG(*)
FROM warehouse
```

```
SELECT @NUM_DISTRICT = COUNT_BIG(*)
FROM district
```

```
SELECT @NUM_ITEM = COUNT_BIG(*)
FROM item
```

```
SELECT @NUM_CUSTOMER = COUNT_BIG(*)
FROM customer
```

```
SELECT @NUM_ORDERS = COUNT_BIG(*)
FROM orders
```

```
SELECT @NUM_ORDER_LINE = COUNT_BIG(*)
FROM order_line
```

```
SELECT @NUM_NEW_ORDER = COUNT_BIG(*)
FROM new_order
```

```
SELECT @NUM_HISTORY = COUNT_BIG(*)
FROM history
```

```
SELECT @NUM_STOCK = COUNT_BIG(*)
FROM stock
```

--- now calculate and set the target values

```
SELECT @WAREHOUSE_TARGET = @NUM_WAREHOUSE,
       @DISTRICT_TARGET   = @NUM_WAREHOUSE * 10,
       @ITEM_TARGET       = 100000,
       @CUSTOMER_TARGET   = @NUM_WAREHOUSE * 30000,
       @ORDERS_TARGET     = @NUM_WAREHOUSE * 30000,
       @ORDERS_TARGET_LOW = @ORDERS_TARGET - FLOOR(@ORDERS_TARGET
* .01),
       @ORDERS_TARGET_HIGH = @ORDERS_TARGET + FLOOR(@ORDERS_TARGET
* .01),
       @ORDER_LINE_TARGET = @NUM_WAREHOUSE * 300000,
       @ORDER_LINE_TARGET_LOW = @ORDER_LINE_TARGET -
FLOOR(@ORDER_LINE_TARGET * .01),
       @ORDER_LINE_TARGET_HIGH = @ORDER_LINE_TARGET +
FLOOR(@ORDER_LINE_TARGET * .01),
       @NEW_ORDER_TARGET = @NUM_WAREHOUSE * 9000,
       @NEW_ORDER_TARGET_LOW = @NEW_ORDER_TARGET -
FLOOR(@NEW_ORDER_TARGET * .01),
       @NEW_ORDER_TARGET_HIGH = @NEW_ORDER_TARGET +
FLOOR(@NEW_ORDER_TARGET * .01),
       @HISTORY_TARGET = @NUM_WAREHOUSE * 30000,
       @STOCK_TARGET = @NUM_WAREHOUSE * 100000
```

```

--- insert the values into TPCC_INFO
INSERT INTO TPCC_INFO VALUES (GETDATE(),
    @NUM_WAREHOUSE,
    @WAREHOUSE_TARGET,
    @NUM_DISTRICT,
    @DISTRICT_TARGET,
    @NUM_ITEM,
    @ITEM_TARGET,
    @NUM_CUSTOMER,
    @CUSTOMER_TARGET,
    @NUM_ORDERS,
    @ORDERS_TARGET,
    @ORDERS_TARGET_LOW,
    @ORDERS_TARGET_HIGH,
    @NUM_ORDER_LINE,
    @ORDER_LINE_TARGET,
    @ORDER_LINE_TARGET_LOW,
    @ORDER_LINE_TARGET_HIGH,
    @NUM_NEW_ORDER,
    @NEW_ORDER_TARGET,
    @NEW_ORDER_TARGET_LOW,
    @NEW_ORDER_TARGET_HIGH,
    @NUM_HISTORY,
    @HISTORY_TARGET,
    @NUM_STOCK,
    @STOCK_TARGET)

```

GO

```

--- output the row counts from the build

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| WAREHOUSE TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_WAREHOUSE AS 'Warehouse Rows',
    WAREHOUSE_TARGET AS 'Warehouse Target',
    CASE WHEN (NUM_WAREHOUSE = WAREHOUSE_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END AS 'Warehouse Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| DISTRICT TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_DISTRICT AS 'District Rows',

```

```

DISTRICT_TARGET          AS 'District Target',
CASE WHEN (NUM_DISTRICT = DISTRICT_TARGET)
  THEN 'OK!'
  ELSE 'ERROR!!!'
END                      AS 'District Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '|  ITEM TABLE  |'
PRINT '-----'

```

```

SELECT TOP 1
  CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
  NUM_ITEM                      AS 'Item Rows',
  ITEM_TARGET                   AS 'Item Target',
CASE WHEN (NUM_ITEM = ITEM_TARGET)
  THEN 'OK!'
  ELSE 'ERROR!!!'
END                              AS 'Item Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '|  CUSTOMER TABLE  |'
PRINT '-----'

```

```

SELECT TOP 1
  CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
  NUM_CUSTOMER                   AS 'Customer Rows',
  CUSTOMER_TARGET                AS 'Customer Target',
CASE WHEN (NUM_CUSTOMER = CUSTOMER_TARGET)
  THEN 'OK!'
  ELSE 'ERROR!!!'
END                              AS 'Customer Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '|  ORDERS TABLE  |'
PRINT '-----'

```

```

SELECT TOP 1
  CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
  NUM_ORDERS                     AS 'Orders Rows',
  ORDERS_TARGET                 AS 'Orders Target',
CASE WHEN (NUM_ORDERS = ORDERS_TARGET)
  THEN 'OK!'
  WHEN (NUM_ORDERS BETWEEN ORDERS_TARGET_LOW AND
ORDERS_TARGET_HIGH)

```

```

        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END          AS 'Orders Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| ORDER LINE TABLE   |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_ORDER_LINE          AS 'Order Line Rows',
    ORDER_LINE_TARGET      AS 'Order Line Target',
    CASE WHEN (NUM_ORDER_LINE = ORDER_LINE_TARGET)
        THEN 'OK!'
        WHEN (NUM_ORDER_LINE BETWEEN ORDER_LINE_TARGET_LOW AND
ORDER_LINE_TARGET_HIGH)
        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END          AS 'Orders Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| NEW ORDER TABLE     |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_NEW_ORDER          AS 'New Order Rows',
    NEW_ORDER_TARGET      AS 'New Order Target',
    CASE WHEN (NUM_NEW_ORDER = NEW_ORDER_TARGET)
        THEN 'OK!'
        WHEN (NUM_NEW_ORDER BETWEEN NEW_ORDER_TARGET_LOW AND
NEW_ORDER_TARGET_HIGH)
        THEN 'OK! (within 1%)'
        ELSE 'ERROR!!!'
    END          AS 'New Order Message'
FROM TPCC_INFO
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| HISTORY TABLE      |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_HISTORY          AS 'History Rows',
    HISTORY_TARGET      AS 'History Target',

```

```

        CASE WHEN (NUM_HISTORY = HISTORY_TARGET)
            THEN 'OK!'
            ELSE 'ERROR!!!'
        END          AS 'New Order Message'
FROM TPCC_INFO
GO

PRINT "
PRINT "
PRINT '-----'
PRINT '| STOCK TABLE |'
PRINT '-----'
SELECT TOP 1
    CONVERT(CHAR(30),INFO_DATE,21) AS 'Date',
    NUM_STOCK                      AS 'Stock Rows',
    STOCK_TARGET                   AS 'Stock Target',
    CASE WHEN (NUM_STOCK = STOCK_TARGET)
        THEN 'OK!'
        ELSE 'ERROR!!!'
    END          AS 'Stock Message'
FROM TPCC_INFO
GO

```

```

-----
-- Check Indexes
-----

```

```

USE tpcc
GO

```

```

PRINT "
PRINT "
PRINT '-----'
PRINT '| TPC-C INDEXES |'
PRINT '-----'
EXEC sp_helpindex warehouse
EXEC sp_helpindex district
EXEC sp_helpindex item
EXEC sp_helpindex customer
EXEC sp_helpindex orders
EXEC sp_helpindex order_line
EXEC sp_helpindex new_order
EXEC sp_helpindex history
EXEC sp_helpindex stock
GO

```

version.sql

```

-----
--
-- File: VERSION.SQL
-- Microsoft TPC-C Benchmark Kit Ver. 4.63
-- Copyright Microsoft, 2005
--

```

```
--          --
--      Extracts current version of SQL Server      --
--          --
-----
USE master
GO

SELECT CONVERT(char(20), SERVERPROPERTY('ProductVersion')),
        CONVERT(char(20), SERVERPROPERTY('ProductLevel')),
        CONVERT(char(29), SERVERPROPERTY('Edition'))
GO

SELECT CONVERT(char(30), GETDATE(), 21)
GO
```


Appendix C: TUNABLE PARAMETERS

Microsoft SQL Server 2005 Configuration Parameters

1> 2> use tpcc

1> 2> sp_configure

name	minimum	maximum	config_value
run_value			

Ad Hoc Distributed Queries		0 1	0
0			
affinity I/O mask	-2147483648	2147483647	0
0			
affinity mask	-2147483648	2147483647	255
255			
affinity64 I/O mask	-2147483648	2147483647	0
0			
affinity64 mask	-2147483648	2147483647	0
0			
Agent XPs	0	1	0
0			
allow updates	0	1	0
0			
awe enabled	0	1	1
1			
blocked process threshold	0	86400	0
0			
c2 audit mode	0	1	0
0			
clr enabled	0	1	0
0			
cost threshold for parallelism	0	32767	5
5			
cross db ownership chaining	0	1	0
0			
cursor threshold	-1	2147483647	-1
-1			
Database Mail XPs	0	1	0
0			
default full-text language	0	2147483647	1033
1033			
default language	0	9999	0
0			
default trace enabled	0	1	0
0			
disallow results from triggers	0	1	0
0			
fill factor (%)	0	100	0
0			
ft crawl bandwidth (max)	0	32767	100
100			

ft crawl bandwidth (min)	0	32767	0
0			
ft notify bandwidth (max)	0	32767	100
100			
ft notify bandwidth (min)	0	32767	0
0			
in-doubt xact resolution	0	2	0
0			
index create memory (KB)	704	2147483647	0
0			
lightweight pooling	0	1	1
1			
locks	5000	2147483647	0
0			
max degree of parallelism	0	64	1
1			
max full-text crawl range	0	256	4
4			
max server memory (MB)	16	2147483647	31750
31750			
max text repl size (B)	0	2147483647	65536
65536			
max worker threads	128	32767	450
450			
media retention	0	365	0
0			
min memory per query (KB)	512	2147483647	1024
1024			
min server memory (MB)	0	2147483647	0
16			
nested triggers	0	1	1
1			
network packet size (B)	512	32767	4096
4096			
Ole Automation Procedures	0	1	0
0			
open objects	0	2147483647	0
0			
PH timeout (s)	1	3600	60
60			
precompute rank	0	1	0
0			
priority boost	0	1	1
1			
query governor cost limit	0	2147483647	0
0			
query wait (s)	-1	2147483647	-1
-1			
recovery interval (min)	0	32767	116
116			
remote access	0	1	1
1			
remote admin connections	0	1	0

```

0
remote login timeout (s)      0 2147483647      20
20
remote proc trans            0      1      0
0
remote query timeout (s)    0 2147483647      0
0
Replication XPs             0      1      0
0
scan for startup procs      0      1      0
0
server trigger recursion    0      1      1
1
set working set size        0      1      1
1
show advanced options       0      1      1
1
SMO and DMO XPs             0      1      1
1
SQL Mail XPs                0      1      0
0
transform noise words       0      1      0
0
two digit year cutoff       1753   9999   2049
2049
user connections            0  32767   0
0
user options                0  32767   0
0
Web Assistant Procedures    0      1      0
0
xp_cmdshell                 0      1      0
0

```

1>

Microsoft Windows Server 2003 Enterprise Edition SP1

Changes to the SUT

Changes made to the default installation of Windows Server 2003 Enterprise

- All services were left in their default setup.

c:\boot.ini

```

[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(3)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(3)\WINDOWS="Windows Server 2003 Enterprise x64 Edition"
/noexecute=optout /fastdetect
multi(0)disk(0)rdisk(0)partition(2)\WINDOWS="Windows Server 2003, Enterprise" /fastdetect

```

```
/NoExecute=OptOut
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="EM64T - Windows Server 2003 Enterprise"
/fastdetect
```

gpedit.msc - Computer Configuration - Windows Settings - Security Settings - Local Policies - User Rights Assignments - policy 'Lock pages in memory' added group 'Administrators'

Microsoft SQL Server 2005 Startup Parameters

```
sqlservr -c -x -T834 -T652 -T8016 -T820 -T3502 -T405 -T661 -T8744 -T888 -T8020 -T677 -T8011 -
T8012 -T8018 -T8019 -T205
```

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
- T *trace#* Indicates that an instance of SQL Server should be started with a specified trace flag (*trace#*) in effect.

Registry

Windows Registry Editor Version 5.00

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect]
"ErrorControl"=dword:00000001
"Type"=dword:00000001
"Start"=dword:00000002
"ImagePath"=hex(2):73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,44,00,\
52,00,49,00,56,00,45,00,52,00,53,00,5c,00,71,00,6c,00,64,00,69,00,72,00,65,\
00,63,00,74,00,2e,00,73,00,79,00,73,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Parameters]
"MaxPathsPerDevice"=dword:00000001
"SrbListSize"=dword:00000400
"PerCpuData"=dword:00000001
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Security]
"Security"=hex:01,00,14,80,b8,00,00,00,c4,00,00,00,14,00,00,00,30,00,00,02,\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,01,00,00,\
00,00,02,00,88,00,06,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,\
05,12,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,05,20,00,00,\
20,02,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,05,04,00,00,00,\
00,14,00,8d,01,02,00,01,01,00,00,00,00,05,06,00,00,00,00,14,00,00,01,\
00,00,01,01,00,00,00,00,05,0b,00,00,00,00,18,00,fd,01,02,00,01,02,00,\
00,00,00,05,20,00,00,00,23,02,00,00,01,01,00,00,00,00,05,12,00,00,\
01,01,00,00,00,00,05,12,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\qldirect\Enum]
"0"="Root\UNKNOWN\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration]

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node0]
"CPUMask"=hex:01,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node1]
"CPUMask"=hex:02,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node2]
"CPUMask"=hex:04,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node3]
"CPUMask"=hex:08,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node4]
"CPUMask"=hex:10,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node5]
"CPUMask"=hex:20,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node6]
"CPUMask"=hex:40,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\90\NodeConfiguration\Node7]
"CPUMask"=hex:80,00,00,00,00,00,00,00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp\IPAll]
"TcpPort"="2001,2002,2003,2004,2005,2006,2007,2008"

SUT System Information Report

System Information report written at: 04/04/06 09:13:41

System Name: TPCCX64

[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003 Enterprise x64 Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	TPCCX64
System Manufacturer	Intel
System Model	Star Lake
System Type	x64-based PC
Processor	EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz

Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 Processor EM64T Family 15 Model 6 Stepping 4 GenuineIntel ~3724 Mhz
 BIOS Version/Date Intel Corporation S5000.86B.01.00.0030.022420061618, 2/24/2006
 SMBIOS Version 2.4
 Windows Directory E:\WINDOWS
 System Directory E:\WINDOWS\system32
 Boot Device \Device\HarddiskVolume123
 Locale United States
 Hardware Abstraction Layer Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)"
 User Name Not Available
 Time Zone China Standard Time
 Total Physical Memory 32,759.75 MB
 Available Physical Memory 30.87 GB
 Total Virtual Memory 33.06 GB
 Available Virtual Memory 32.69 GB
 Page File Space 2.00 GB
 Page File E:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
I/O Port 0x00000000-0x00000CF7	PCI bus
I/O Port 0x00000000-0x00000CF7	Direct memory access controller
IRQ 20	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B
IRQ 20	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
I/O Port 0x00002000-0x00002FFF	Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690
I/O Port 0x00002000-0x00002FFF	QLogic Fibre Channel Adapter
IRQ 10	PCI Serial Port
IRQ 10	Intel(R) 631xESB/6321ESB SMBus Controller - 269B
IRQ 11	PCI Device
IRQ 11	PCI Device
IRQ 23	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688
IRQ 23	Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller - 268C
I/O Port 0x00003000-0x00004FFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
I/O Port 0x00003000-0x00004FFF	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500

I/O Port 0x00003000-0x00004FFF Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 - 3518
 I/O Port 0x00003000-0x00004FFF Standard Universal PCI to USB Host Controller

Memory Address 0xFEC00000-0xFECFFFFFFF Motherboard resources
 Memory Address 0xFEC00000-0xFECFFFFFFF Advanced programmable interrupt controller

Memory Address 0xFE700000-0xFE7003FF Base System Device
 Memory Address 0xFE700000-0xFE7003FF Motherboard resources

Memory Address 0xA8900000-0xA89FFFFFFF Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
 Memory Address 0xA8900000-0xA89FFFFFFF QLogic Fibre Channel Adapter

IRQ 16 Intel(R) 5000P Chipset Memory Controller Hub - 25D8
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7
 IRQ 16 Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500
 IRQ 16 Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
 IRQ 16 QLogic Fibre Channel Adapter
 IRQ 16 Standard Universal PCI to USB Host Controller
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 4 - 25E4
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9
 IRQ 16 Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7
 IRQ 16 Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690
 IRQ 16 QLogic Fibre Channel Adapter

IRQ 17 QLogic Fibre Channel Adapter
 IRQ 17 Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E2 - 3514
 IRQ 17 Standard Dual Channel PCI IDE Controller
 IRQ 17 QLogic Fibre Channel Adapter

IRQ 18 Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 - 3518
 IRQ 18 Intel(R) PRO/1000 EB Network Connection with I/O Acceleration #2

Memory Address 0xA0000-0xBFFFFF PCI bus
 Memory Address 0xA0000-0xBFFFFF Standard VGA Graphics Adapter

Memory Address 0xA8B00000-0xA8BFFFFFFF Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690
 Memory Address 0xA8B00000-0xA8BFFFFFFF QLogic Fibre Channel Adapter

I/O Port 0x00004000-0x00004FFF Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510
 I/O Port 0x00004000-0x00004FFF QLogic Fibre Channel Adapter

Memory Address 0xA0000000-0xFE000000 PCI bus
 Memory Address 0xA0000000-0xFE000000 Standard VGA Graphics Adapter

IRQ 1 Base System Device
 IRQ 1 Standard 101/102-Key or Microsoft Natural PS/2 Keyboard

Memory Address 0xA8000000-0xA89FFFFF 2-3 - 25F7	Intel(R) 5000 Series Chipset PCI Express x8 Port
Memory Address 0xA8000000-0xA89FFFFF Upstream Port - 3500	Intel(R) 6311ESB/6321ESB PCI Express
Memory Address 0xA8000000-0xA89FFFFF Downstream Port E3 - 3518	Intel(R) 6311ESB/6321ESB PCI Express
Memory Address 0xA8000000-0xA89FFFFF I/O Acceleration	Intel(R) PRO/1000 EB Network Connection with

[DMA]

Resource	Device	Status
Channel 4	Direct memory access controller	OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource	Device	Status
0x00000000-0x00000CF7	PCI bus	OK
0x00000000-0x00000CF7	Direct memory access controller	OK
0x00000D00-0x0000FFFF	PCI bus	OK
0x00003000-0x00004FFF OK	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	
0x00003000-0x00004FFF OK	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500	
0x00003000-0x00004FFF 3518 OK	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 -	
0x00003000-0x00004FFF	Standard Universal PCI to USB Host Controller	OK
0x00004000-0x00004FFF 3510 OK	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 -	
0x00004000-0x00004FFF	QLogic Fibre Channel Adapter	OK
0x00004400-0x000044FF	QLogic Fibre Channel Adapter	OK
0x00003040-0x0000305F #2 OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration	
0x00003020-0x0000303F OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration	
0x00003080-0x00003087	Standard Dual Channel PCI IDE Controller	OK
0x00003094-0x00003097	Standard Dual Channel PCI IDE Controller	OK
0x00003078-0x0000307F	Standard Dual Channel PCI IDE Controller	OK
0x00003090-0x00003093	Standard Dual Channel PCI IDE Controller	OK
0x00003060-0x0000306F	Standard Dual Channel PCI IDE Controller	OK
0x00003070-0x00003077	PCI Serial Port	OK
0x0000308C-0x0000308F	PCI Device	OK
0x00003088-0x0000308B	PCI Device	OK
0x00002000-0x00002FFF OK	Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690	
0x00002000-0x00002FFF	QLogic Fibre Channel Adapter	OK
0x00002400-0x000024FF	QLogic Fibre Channel Adapter	OK
0x00005080-0x0000509F	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688	

OK	
0x00005060-0x0000507F	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2689
OK	
0x00005040-0x0000505F	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268A
OK	
0x00005020-0x0000503F	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B
OK	
0x00001000-0x000010FF	Standard VGA Graphics Adapter OK
0x000003B0-0x000003BB	Standard VGA Graphics Adapter OK
0x000003C0-0x000003DF	Standard VGA Graphics Adapter OK
0x00000081-0x00000083	Direct memory access controller OK
0x00000087-0x00000087	Direct memory access controller OK
0x00000089-0x0000008B	Direct memory access controller OK
0x0000008F-0x0000008F	Direct memory access controller OK
0x000000C0-0x000000DF	Direct memory access controller OK
0x00000070-0x00000071	System CMOS/real time clock OK
0x00000074-0x00000077	System CMOS/real time clock OK
0x00000020-0x0000003D	Programmable interrupt controller OK
0x000000A0-0x000000BD	Programmable interrupt controller OK
0x000004D0-0x000004D1	Programmable interrupt controller OK
0x000000F0-0x000000FF	Numeric data processor OK
0x00000040-0x00000043	System timer OK
0x00000050-0x00000053	System timer OK
0x00000061-0x00000061	System speaker OK
0x00000500-0x0000053F	Motherboard resources OK
0x00000400-0x0000047F	Motherboard resources OK
0x00000092-0x00000092	Motherboard resources OK
0x00000800-0x0000087F	Motherboard resources OK
0x00000010-0x0000001F	Motherboard resources OK
0x00000072-0x00000073	Motherboard resources OK
0x00000080-0x00000080	Motherboard resources OK
0x00000084-0x00000086	Motherboard resources OK
0x00000088-0x00000088	Motherboard resources OK
0x0000008C-0x0000008E	Motherboard resources OK
0x00000090-0x0000009F	Motherboard resources OK
0x00000CA0-0x00000CAF	Motherboard resources OK
0x00000060-0x00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard OK
0x000003F8-0x000003FF	Communications Port (COM1) OK
0x000002F8-0x000002FF	Communications Port (COM2) OK
0x000050B8-0x000050BF	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
OK	
0x000050C4-0x000050C7	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
OK	
0x000050B0-0x000050B7	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
OK	
0x000050C0-0x000050C3	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
OK	
0x000050A0-0x000050AF	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
OK	
0x00005000-0x0000501F	Intel(R) 631xESB/6321ESB SMBus Controller - 269B OK

[IRQs]

Resource	Device	Status
IRQ 9	Microsoft ACPI-Compliant System	OK
IRQ 16	Intel(R) 5000P Chipset Memory Controller Hub - 25D8	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	OK
IRQ 16	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500	OK
IRQ 16	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510	OK
IRQ 16	QLogic Fibre Channel Adapter	OK
IRQ 16	Standard Universal PCI to USB Host Controller	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 4 - 25E4	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9	OK
IRQ 16	Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7	OK
IRQ 16	Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690	OK
IRQ 16	QLogic Fibre Channel Adapter	OK
IRQ 17	QLogic Fibre Channel Adapter	OK
IRQ 17	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E2 - 3514	OK
IRQ 17	Standard Dual Channel PCI IDE Controller	OK
IRQ 17	QLogic Fibre Channel Adapter	OK
IRQ 18	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 - 3518	OK
IRQ 18	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration #2	OK
IRQ 19	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration	OK
IRQ 10	PCI Serial Port	OK
IRQ 10	Intel(R) 631xESB/6321ESB SMBus Controller - 269B	OK
IRQ 11	PCI Device	OK
IRQ 11	PCI Device	OK
IRQ 1	Base System Device	OK
IRQ 1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
IRQ 23	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688	OK
IRQ 23	Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller - 268C	OK
IRQ 22	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2689	OK
IRQ 21	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268A	OK
IRQ 20	Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B	OK
IRQ 20	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680	OK
IRQ 8	System CMOS/real time clock	OK
IRQ 13	Numeric data processor	OK
IRQ 0	System timer	OK
IRQ 12	Microsoft PS/2 Mouse	OK
IRQ 4	Communications Port (COM1)	OK
IRQ 3	Communications Port (COM2)	OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	Standard VGA Graphics Adapter	OK
0xA0000000-0xFE000000	PCI bus	OK
0xA0000000-0xFE000000	Standard VGA Graphics Adapter	OK
0xA8000000-0xA89FFFFF	Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7	OK
0xA8000000-0xA89FFFFF	Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500	OK

0xA8000000-0xA89FFFFFF 3518 OK	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 -
0xA8000000-0xA89FFFFFF OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
0xA8900000-0xA89FFFFFF 3510 OK	Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 -
0xA8900000-0xA89FFFFFF	QLogic Fibre Channel Adapter OK
0xA8904000-0xA8907FFF	QLogic Fibre Channel Adapter OK
0xA8820000-0xA883FFFF #2 OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
0xA8400000-0xA87FFFFFF #2 OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
0xA8800000-0xA881FFFF OK	Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
0xA8843000-0xA8843FFF	PCI Serial Port OK
0xA8842000-0xA8842FFF	PCI Device OK
0xA8841000-0xA8841FFF	Standard Universal PCI to USB Host Controller OK
0xA8840000-0xA8840FFF	PCI Device OK
0xFE700000-0xFE7003FF	Base System Device OK
0xFE700000-0xFE7003FF	Motherboard resources OK
0xA8B00000-0xA8BFFFFFF OK	Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690
0xA8B00000-0xA8BFFFFFF	QLogic Fibre Channel Adapter OK
0xA8B04000-0xA8B07FFF	QLogic Fibre Channel Adapter OK
0xA8C00400-0xA8C007FF 268C OK	Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller -
0xA8A00000-0xA8A0FFFF	Standard VGA Graphics Adapter OK
0xFED00000-0xFED003FF	High precision event timer OK
0xA8C00000-0xA8C003FF OK	Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
0xD0000000-0xDFFFFFFF	Motherboard resources OK
0xFED1C000-0xFED1FFFF	Motherboard resources OK
0xFFC00000-0xFFFFFFFF	Motherboard resources OK
0xFEC00000-0xFECFFFFFF	Motherboard resources OK
0xFEC00000-0xFECFFFFFF	Advanced programmable interrupt controller OK
0xFEE00000-0xFEEFFFFFF	Motherboard resources OK
0xFE600000-0xFE6FFFFF	Motherboard resources OK
0xFE000000-0xFE01FFFF	Motherboard resources OK
0x0000-0x9FFFF	Motherboard resources OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
e:\windows\system32\msadp32.acm	Microsoft Corporation						OK
	E:\WINDOWS\system32\MSADP32.ACM				5.2.3790.1830		(srv03_sp1_rtm.050324-

1447) 23.50 KB (24,064 bytes) 3/25/2005 8:00 PM
 e:\windows\system32\msg711.acm Microsoft Corporation OK
 E:\WINDOWS\system32\MSG711.ACM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 13.50 KB (13,824 bytes) 3/25/2005 8:00 PM
 e:\windows\system32\msgsm32.acm Microsoft Corporation OK
 E:\WINDOWS\system32\MSGSM32.ACM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 34.50 KB (35,328 bytes) 3/25/2005 8:00 PM
 e:\windows\system32\tssoft32.acm DSP GROUP, INC. OK
 E:\WINDOWS\system32\TSSOFT32.ACM 1.01 13.50 KB (13,824 bytes)
 3/25/2005 8:00 PM
 e:\windows\system32\imaadp32.acm Microsoft Corporation OK
 E:\WINDOWS\system32\IMAADP32.ACM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 24.00 KB (24,576 bytes) 3/25/2005 8:00 PM

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
e:\windows\system32\msrle32.dll	Microsoft Corporation		OK				
		E:\WINDOWS\system32\MSRLE32.DLL		5.2.3790.1830		(srv03_sp1_rtm.050324-1447)	
		15.50 KB (15,872 bytes)		3/25/2005 8:00 PM			
e:\windows\system32\tsbyuv.dll	Microsoft Corporation		OK				
		E:\WINDOWS\system32\TSBYUV.DLL		5.2.3790.1830		(srv03_sp1_rtm.050324-1447)	
		12.50 KB (12,800 bytes)		3/25/2005 1:34 AM			
e:\windows\system32\msyuv.dll	Microsoft Corporation		OK				
		E:\WINDOWS\system32\MSYUV.DLL		5.2.3790.1830		(srv03_sp1_rtm.050324-1447)	
		21.00 KB (21,504 bytes)		3/25/2005 1:21 AM			
e:\windows\system32\iyuv_32.dll	Microsoft Corporation		OK				
		E:\WINDOWS\system32\IYUV_32.DLL		5.2.3790.1830		(srv03_sp1_rtm.050324-1447)	
		52.50 KB (53,760 bytes)		3/25/2005 1:19 AM			
e:\windows\system32\msvidc32.dll	Microsoft Corporation		OK				
		E:\WINDOWS\system32\MSVIDC32.DLL		5.2.3790.1830		(srv03_sp1_rtm.050324-1447)	
		43.00 KB (44,032 bytes)		3/25/2005 8:00 PM			

[CD-ROM]

Item Value

[Sound Device]

Item Value

[Display]

Item	Value	ID
Name	Standard VGA Graphics Adapter	
PNP	PCI\VEN_1002&DEV_515E&SUBSYS_34788086&REV_02\4&2014205D&0&60F0	
Adapter Type	ATI ES1000, (Standard display types) compatible	
Adapter Description	Standard VGA Graphics Adapter	
Adapter RAM	16.00 MB (16,777,216 bytes)	
Installed Drivers	vga.dll,framebuf.dll,vga256,vga64k	
Driver Version	5.2.3790.1830	
INF File	display.inf (vga section)	

Color Planes 1
Color Table Entries 4294967296
Resolution 1024 x 768 x 1 hertz
Bits/Pixel 32
Memory Address 0xA0000000-0xFE000000
I/O Port 0x00001000-0x000010FF
Memory Address 0xA8A00000-0xA8A0FFFF
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver e:\windows\system32\drivers\vgapnp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
33.00 KB (33,792 bytes), 3/30/2006 6:18 PM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&2AA4AD3D&0
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver e:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
91.00 KB (93,184 bytes), 3/25/2005 8:00 PM)

[Pointing Device]

Item Value
Hardware Type Microsoft PS/2 Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F03\4&2AA4AD3D&0
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver e:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
91.00 KB (93,184 bytes), 3/25/2005 8:00 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item	Value
Name	[00000001] RAS Async Adapter
Adapter Type	Not Available
Product Type	RAS Async Adapter
Installed	Yes
PNP Device ID	Not Available
Last Reset	4/3/2006 12:56 PM
Index	1
Service Name	AsyncMac
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	No
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available

Name	[00000002] WAN Miniport (L2TP)
Adapter Type	Not Available
Product Type	WAN Miniport (L2TP)
Installed	Yes
PNP Device ID	ROOT\MS_L2TPMINIPOINT\0000
Last Reset	4/3/2006 12:56 PM
Index	2
Service Name	Rasl2tp
IP Address	Not Available
IP Subnet	Not Available
Default IP Gateway	Not Available
DHCP Enabled	No
DHCP Server	Not Available
DHCP Lease Expires	Not Available
DHCP Lease Obtained	Not Available
MAC Address	Not Available
Driver	e:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 132.00 KB (135,168 bytes), 3/25/2005 8:00 PM)

Name	[00000003] WAN Miniport (PPTP)
Adapter Type	Wide Area Network (WAN)
Product Type	WAN Miniport (PPTP)
Installed	Yes
PNP Device ID	ROOT\MS_PPTPMINIPOINT\0000
Last Reset	4/3/2006 12:56 PM
Index	3
Service Name	PptpMiniport
IP Address	Not Available
IP Subnet	Not Available

Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver e:\windows\system32\drivers\raspptp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 117.50 KB (120,320 bytes), 3/25/2005 8:00 PM)

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPOINT\0000
Last Reset 4/3/2006 12:56 PM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver e:\windows\system32\drivers\rasppoe.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 67.50 KB (69,120 bytes), 3/25/2005 8:00 PM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 4/3/2006 12:56 PM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver e:\windows\system32\drivers\raspti.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 30.50 KB (31,232 bytes), 3/25/2005 8:00 PM)

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 4/3/2006 12:56 PM

Index 6

Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver e:\windows\system32\drivers\ndiswan.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 157.50 KB (161,280 bytes), 3/25/2005 8:00 PM)

Name [00000007] Intel(R) PRO/1000 EB Network Connection with I/O Acceleration

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_1096&SUBSYS_34768086&REV_01\6&AA0C0A6&0&01100010
Last Reset 4/3/2006 12:56 PM

Index 7

Service Name e1express
IP Address 0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway Not Available
DHCP Enabled Yes
DHCP Server
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:15:17:01:D9:0F
Memory Address 0xA8800000-0xA881FFFF
Memory Address 0xA8000000-0xA89FFFFFF
I/O Port 0x00003020-0x0000303F
IRQ Channel IRQ 19
Driver e:\windows\system32\drivers\e1e5132e.sys (9.3.28.0 built by: WinDDK, 216.50 KB (221,696 bytes), 1/1/2001 4:00 AM)

Name [00000008] Intel(R) PRO/1000 EB Network Connection with I/O Acceleration

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 EB Network Connection with I/O Acceleration
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_1096&SUBSYS_34768086&REV_01\6&AA0C0A6&0&00100010
Last Reset 4/3/2006 12:56 PM

Index 8

Service Name e1express
IP Address 192.168.0.201
IP Subnet 255.255.255.0
Default IP Gateway 192.168.0.254
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available

MAC Address 00:15:17:01:D9:0E
Memory Address 0xA8820000-0xA883FFFF
Memory Address 0xA8400000-0xA87FFFFF
I/O Port 0x00003040-0x0000305F
IRQ Channel IRQ 18
Driver e:\windows\system32\drivers\le5132e.sys (9.3.28.0 built by: WinDDK, 216.50 KB (221,696 bytes), 1/1/2001 4:00 AM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)

Message Oriented Yes
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item Value
File e:\windows\system32\wsock32.dll
Size 24.50 KB (25,088 bytes)
Version 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)

[Ports]

[Serial]

Item Value
Name Communications Port (COM1)
Status OK
PNP Device ID ACPI\PNP0501\1
Maximum Input Buffer Size 0
Maximum Output Buffer Size No
Settable Baud Rate Yes
Settable Data Bits Yes
Settable Flow Control Yes
Settable Parity Yes

Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000003F8-0x000003FF
 IRQ Channel IRQ 4
 Driver e:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
 118.50 KB (121,344 bytes), 3/25/2005 8:00 PM)

Name Communications Port (COM2)
 Status OK
 PNP Device ID ACPI\PNP0501\2
 Maximum Input Buffer Size 0
 Maximum Output Buffer Size No
 Settable Baud Rate Yes
 Settable Data Bits Yes
 Settable Flow Control Yes
 Settable Parity Yes
 Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8

Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000002F8-0x000002FF
 IRQ Channel IRQ 3
 Driver e:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
 118.50 KB (121,344 bytes), 3/25/2005 8:00 PM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value

Drive B:
 Description Network Connection
 Provider Name \\192.168.0.161\mykit

Drive C:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 29.29 GB (31,453,437,952 bytes)
 Free Space 27.19 GB (29,197,000,704 bytes)
 Volume Name
 Volume Serial Number CC4AAD1D

Drive D:

Description Local Fixed Disk
Compressed No
File System NTFS
Size 29.29 GB (31,453,437,952 bytes)
Free Space 26.89 GB (28,873,560,064 bytes)
Volume Name
Volume Serial Number 60E94757

Drive E:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 53.16 GB (57,083,408,384 bytes)
Free Space 28.99 GB (31,131,205,632 bytes)
Volume Name
Volume Serial Number 002BC7D9

Drive H:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.48 GB (54,200,115,200 bytes)
Volume Name New Volume
Volume Serial Number 4C46843E

Drive I:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 256.20 GB (275,095,486,464 bytes)
Volume Name New Volume
Volume Serial Number 9860872A

Drive J:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.48 GB (54,200,254,464 bytes)
Volume Name New Volume
Volume Serial Number D0612BCA

Drive K:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.48 GB (54,200,254,464 bytes)
Volume Name New Volume
Volume Serial Number 806DDD69

Drive L:
Description Local Fixed Disk
Compressed Not Available
File System Not Available
Size Not Available
Free Space Not Available
Volume Name Not Available
Volume Serial Number Not Available

Drive M:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.47 GB (54,191,726,592 bytes)
Volume Name New Volume
Volume Serial Number DCD3C3C3

Drive N:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.47 GB (54,191,861,760 bytes)
Volume Name New Volume
Volume Serial Number 44E36B7C

Drive R:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.47 GB (54,191,861,760 bytes)
Volume Name New Volume
Volume Serial Number F4FB0072

Drive S:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 50.47 GB (54,191,861,760 bytes)
Volume Name New Volume
Volume Serial Number 74FF636D

Drive U:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 443.13 GB (475,810,852,864 bytes)
Volume Name New Volume
Volume Serial Number 5058A7AA

Drive W:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 256.20 GB (275,095,396,352 bytes)
Volume Name New Volume
Volume Serial Number 8847B160

Drive X:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 256.20 GB (275,095,465,984 bytes)
Volume Name New Volume
Volume Serial Number 5050BC6C

Drive Y:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 256.20 GB (275,095,465,984 bytes)
Volume Name New Volume
Volume Serial Number A458ABDE

Drive Z:
Description Local Fixed Disk
Compressed No
File System NTFS
Size 443.31 GB (476,005,175,296 bytes)
Free Space 256.20 GB (275,095,465,984 bytes)
Volume Name New Volume
Volume Serial Number 54610663

[Disks]

Item	Value
Description	Disk drive
Manufacturer	(Standard disk drives)
Model	Maxtor 6Y120M0
Bytes/Sector	512
Media Loaded	Yes
Media Type	Fixed hard disk
Partitions	3
SCSI Bus	0
SCSI Logical Unit	0
SCSI Port	2
SCSI Target ID	0
Sectors/Track	63
Size	111.76 GB (119,998,609,920 bytes)

Total Cylinders 14,589
Total Sectors 234,372,285
Total Tracks 3,720,195
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 29.29 GB (31,453,438,464 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #0, Partition #1
Partition Size 82.46 GB (88,536,913,920 bytes)
Partition Starting Offset 31,453,470,720 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 0
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #12, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #12, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #12, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 1
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880

Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #13, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #13, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #13, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 2
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #14, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #14, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #14, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 3
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952

Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #15, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #15, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #15, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 4
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #16, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #16, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #16, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 5
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)

Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #17, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #17, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #17, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 5
SCSI Target ID 6
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #18, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #18, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #18, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 0
Sectors/Track 63

Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #26, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #26, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #26, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 1
Sectors/Track 63

Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #27, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #27, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #27, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #27, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0

SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 2
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #28, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #28, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #28, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 3
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #29, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #29, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #29, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #29, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512

Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 7
 SCSI Target ID 4
 Sectors/Track 63
 Size 474.58 GB (509,572,546,560 bytes)
 Total Cylinders 61,952
 Total Sectors 995,258,880
 Total Tracks 15,797,760
 Tracks/Cylinder 255
 Partition Disk #30, Partition #0
 Partition Size 1,004.03 MB (1,052,803,584 bytes)
 Partition Starting Offset 32,256 bytes
 Partition Disk #30, Partition #1
 Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #30, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model EUROLOGC FC2502 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 4
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 7
 SCSI Target ID 5
 Sectors/Track 63
 Size 474.58 GB (509,572,546,560 bytes)
 Total Cylinders 61,952
 Total Sectors 995,258,880
 Total Tracks 15,797,760
 Tracks/Cylinder 255
 Partition Disk #31, Partition #0
 Partition Size 1,004.03 MB (1,052,803,584 bytes)
 Partition Starting Offset 32,256 bytes
 Partition Disk #31, Partition #1
 Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #31, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes
 Partition Disk #31, Partition #3
 Partition Size 443.31 GB (476,005,178,880 bytes)
 Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 6
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #32, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #32, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #32, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 7
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #33, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #33, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #33, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Partition Disk #33, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 8
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255

Partition Disk #34, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #34, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #34, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 9
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #35, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #35, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)

Partition Starting Offset 1,052,835,840 bytes
Partition Disk #35, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 10
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #36, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #36, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #36, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 7
SCSI Target ID 11
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #37, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #37, Partition #1

Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #37, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 0
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #19, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #19, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #19, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 1
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #20, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes

Partition Disk #20, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #20, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 2
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255

Partition Disk #21, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #21, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #21, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 3
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255

Partition Disk #22, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)

Partition Starting Offset 32,256 bytes
Partition Disk #22, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #22, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 4
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #23, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #23, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #23, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 5
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #24, Partition #0

Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #24, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #24, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 1
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 6
SCSI Target ID 6
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #25, Partition #0
Partition Size 474.58 GB (509,572,514,304 bytes)
Partition Starting Offset 32,256 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 0
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #1, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #1, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes

Partition Disk #1, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #1, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 1
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255

Partition Disk #2, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #2, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #2, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #2, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 2
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760

Tracks/Cylinder 255
Partition Disk #3, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #3, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #3, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #3, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 3
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #4, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #4, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #4, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #4, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0

SCSI Port 4
SCSI Target ID 4
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #5, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #5, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #5, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 5
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #6, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #6, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #6, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #6, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes

Media Type Fixed hard disk
 Partitions 4
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 4
 SCSI Target ID 6
 Sectors/Track 63
 Size 474.58 GB (509,572,546,560 bytes)
 Total Cylinders 61,952
 Total Sectors 995,258,880
 Total Tracks 15,797,760
 Tracks/Cylinder 255
 Partition Disk #7, Partition #0
 Partition Size 1,004.03 MB (1,052,803,584 bytes)
 Partition Starting Offset 32,256 bytes
 Partition Disk #7, Partition #1
 Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #7, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes
 Partition Disk #7, Partition #3
 Partition Size 443.31 GB (476,005,178,880 bytes)
 Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model EUROLOGC FC2502 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 4
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 4
 SCSI Target ID 7
 Sectors/Track 63
 Size 474.58 GB (509,572,546,560 bytes)
 Total Cylinders 61,952
 Total Sectors 995,258,880
 Total Tracks 15,797,760
 Tracks/Cylinder 255
 Partition Disk #8, Partition #0
 Partition Size 1,004.03 MB (1,052,803,584 bytes)
 Partition Starting Offset 32,256 bytes
 Partition Disk #8, Partition #1
 Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #8, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes
 Partition Disk #8, Partition #3
 Partition Size 443.31 GB (476,005,178,880 bytes)

Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 8
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #9, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #9, Partition #1
Partition Size 19.53 GB (20,974,464,000 bytes)
Partition Starting Offset 1,052,835,840 bytes
Partition Disk #9, Partition #2
Partition Size 10.75 GB (11,540,067,840 bytes)
Partition Starting Offset 22,027,299,840 bytes
Partition Disk #9, Partition #3
Partition Size 443.31 GB (476,005,178,880 bytes)
Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
Manufacturer (Standard disk drives)
Model EUROLOGC FC2502 SCSI Disk Device
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 4
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 4
SCSI Target ID 9
Sectors/Track 63
Size 474.58 GB (509,572,546,560 bytes)
Total Cylinders 61,952
Total Sectors 995,258,880
Total Tracks 15,797,760
Tracks/Cylinder 255
Partition Disk #10, Partition #0
Partition Size 1,004.03 MB (1,052,803,584 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #10, Partition #1

Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #10, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes
 Partition Disk #10, Partition #3
 Partition Size 443.31 GB (476,005,178,880 bytes)
 Partition Starting Offset 33,567,367,680 bytes

Description Disk drive
 Manufacturer (Standard disk drives)
 Model EUROLOGC FC2502 SCSI Disk Device
 Bytes/Sector 512
 Media Loaded Yes
 Media Type Fixed hard disk
 Partitions 3
 SCSI Bus 0
 SCSI Logical Unit 0
 SCSI Port 4
 SCSI Target ID 10
 Sectors/Track 63
 Size 474.58 GB (509,572,546,560 bytes)
 Total Cylinders 61,952
 Total Sectors 995,258,880
 Total Tracks 15,797,760
 Tracks/Cylinder 255
 Partition Disk #11, Partition #0
 Partition Size 1,004.03 MB (1,052,803,584 bytes)
 Partition Starting Offset 32,256 bytes
 Partition Disk #11, Partition #1
 Partition Size 19.53 GB (20,974,464,000 bytes)
 Partition Starting Offset 1,052,835,840 bytes
 Partition Disk #11, Partition #2
 Partition Size 10.75 GB (11,540,067,840 bytes)
 Partition Starting Offset 22,027,299,840 bytes

[SCSI]

Item	Value
Name	QLogic Fibre Channel Adapter
Manufacturer	QLogic
Status	OK
PNP	Device ID
	PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\6&AECCA84&0&00000010
I/O Port	0x00004400-0x000044FF
Memory Address	0xA8900000-0xA89FFFFF
IRQ Channel	IRQ 16
Driver	e:\windows\system32\drivers\ql2300.sys (9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes), 3/14/2006 2:33 PM)

Name QLogic Fibre Channel Adapter
 Manufacturer QLogic
 Status OK

PNP Device ID
PCIIVEN_1077&DEV_2432&SUBSYS_01381077&REV_02\6&AECCA84&0&01000010
I/O Port 0x00004000-0x00004FFF
Memory Address 0xA8904000-0xA8907FFF
IRQ Channel IRQ 17
Driver e:\windows\system32\drivers\ql2300.sys (9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes), 3/14/2006 2:33 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK

PNP Device ID
PCIIVEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&187919FE&0&00E0
I/O Port 0x00002400-0x000024FF
Memory Address 0xA8B00000-0xA8BFFFFF
IRQ Channel IRQ 16
Driver e:\windows\system32\drivers\ql2300.sys (9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes), 3/14/2006 2:33 PM)

Name QLogic Fibre Channel Adapter
Manufacturer QLogic
Status OK

PNP Device ID
PCIIVEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&187919FE&0&01E0
I/O Port 0x00002000-0x00002FFF
Memory Address 0xA8B04000-0xA8B07FFF
IRQ Channel IRQ 17
Driver e:\windows\system32\drivers\ql2300.sys (9.1.0.12+MMIO (wx64 IP), 1.12 MB (1,178,624 bytes), 3/14/2006 2:33 PM)

Name QLogic Optimizing and Multipath Driver
Manufacturer QLogic
Status Degraded
PNP Device ID ROOT\UNKNOWN\0000
Driver e:\windows\system32\drivers\qldirect.sys (8.01.13 Beta 3 (wx64), 62.00 KB (63,488 bytes), 3/30/2006 11:16 AM)

[IDE]

Item Value
Name Standard Dual Channel PCI IDE Controller
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK

PNP Device ID
PCI\VEN_8086&DEV_1084&SUBSYS_00008086&REV_01\6&AA0C0A6&0&02100010
I/O Port 0x00003080-0x00003087
I/O Port 0x00003094-0x00003097
I/O Port 0x00003078-0x0000307F
I/O Port 0x00003090-0x00003093
I/O Port 0x00003060-0x0000306F
IRQ Channel IRQ 17
Driver e:\windows\system32\drivers\pciide.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 6.00 KB (6,144 bytes), 3/25/2005 8:00 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\IDE\IDECHANNEL\7&D715D63&0&0
Driver e:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 8:00 PM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\IDE\IDECHANNEL\7&D715D63&0&1
Driver e:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 8:00 PM)

Name Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680
Manufacturer Intel
Status OK
PNP Device ID PCI\VEN_8086&DEV_2680&SUBSYS_34788086&REV_09\3&61AAA01&0&FA
I/O Port 0x000050B8-0x000050BF
I/O Port 0x000050C4-0x000050C7
I/O Port 0x000050B0-0x000050B7
I/O Port 0x000050C0-0x000050C3
I/O Port 0x000050A0-0x000050AF
Memory Address 0xA8C00000-0xA8C003FF
IRQ Channel IRQ 20
Driver e:\windows\system32\drivers\pciide.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 6.00 KB (6,144 bytes), 3/25/2005 8:00 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\IDE\IDECHANNEL\4&6BD4A7B&0&0
Driver e:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 8:00 PM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCI\IDE\IDECHANNEL\4&6BD4A7B&0&1
Driver e:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 145.00 KB (148,480 bytes), 3/25/2005 8:00 PM)

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code	Serial	Port
PCI	PCI\VEN_8086&DEV_1085&SUBSYS_00008086&REV_01\6&AA0C0A6&0&03100010			
	The drivers for this device are not installed.			

PCI Device
 PCIIVEN_8086&DEV_1086&SUBSYS_00008086&REV_01\6&AA0C0A6&0&04100010
 The drivers for this device are not installed.

PCI Device
 PCIIVEN_8086&DEV_1089&SUBSYS_00008086&REV_01\6&AA0C0A6&0&07100010
 The drivers for this device are not installed.

Base System Device
 PCIIVEN_8086&DEV_1A38&SUBSYS_34788086&REV_92\3&61AAA01&0&40
 The drivers for this device are not installed.

[USB]

Device Standard	PNP Universal	Device PCI	PNP to	Device USB	PNP Host	Device Controller
Intel(R) 631xESB/6321ESB	PCIIVEN_8086&DEV_1087&SUBSYS_00008086&REV_01\6&AA0C0A6&0&05100010	USB Universal	Host	Controller	-	2688
Intel(R) 631xESB/6321ESB	PCIIVEN_8086&DEV_2688&SUBSYS_34788086&REV_09\3&61AAA01&0&E8	USB Universal	Host	Controller	-	2689
Intel(R) 631xESB/6321ESB	PCIIVEN_8086&DEV_2689&SUBSYS_34788086&REV_09\3&61AAA01&0&E9	USB Universal	Host	Controller	-	268A
Intel(R) 631xESB/6321ESB	PCIIVEN_8086&DEV_268A&SUBSYS_34788086&REV_09\3&61AAA01&0&EA	USB Universal	Host	Controller	-	268B
Intel(R) 631xESB/6321ESB	PCIIVEN_8086&DEV_268B&SUBSYS_34788086&REV_09\3&61AAA01&0&EB	USB2 Enhanced	Host	Controller	-	268C
	PCIIVEN_8086&DEV_268C&SUBSYS_34788086&REV_09\3&61AAA01&0&EF					

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State	Status	Error	Control
abiosdsk	Abiosdsk	e:\windows\system32\drivers\abiosdsk.sys	Kernel Driver	No	Kernel	Disabled	Stopped		Stopped
acpi	Microsoft ACPI Driver	e:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes	Normal	Running	OK		Yes
acpiec	ACPIEC	e:\windows\system32\drivers\acpiec.sys	Kernel Driver	No	Normal	Disabled	Stopped	OK	No
adpu160m	adpu160m		Kernel Driver	No	Kernel	Disabled	Stopped		Stopped
adpu320	adpu320		Kernel Driver	No	Kernel	Disabled	Stopped		Stopped
afd	AFD	e:\windows\system32\drivers\afd.sys	Kernel Driver	Yes	Normal	Running	OK		System
aic78u2	aic78u2		Kernel Driver	No	Kernel	Disabled	Stopped	OK	Normal
aic78xx	aic78xx		Kernel Driver	No	Kernel	Disabled	Stopped	OK	Normal
aliide	AliIde		Kernel Driver	No	Kernel	Disabled	Stopped	OK	Normal
amdide	AmdIde		Kernel Driver	No	Kernel	Disabled	Stopped	OK	Normal

arc	No	No	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
asynmac	Kernel Driver	No	Manual	Stopped	OK	Normal	No	No	
atapi	Standard IDE/ESDI Hard Disk Controller								
Driver	Yes	Boot	Running	OK	Normal	No	Yes		Kernel
atdisk	Atdisk	No	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore
atmarpc	ATM ARP Client Protocol								
	No	Manual	Stopped	OK	Normal	No	No		
audstub	Audio Stub Driver								
	Manual	Running	OK	Normal	No	Yes			Kernel Driver
beep	Beep								
	Running	OK	Normal	No	Yes				Kernel Driver
cdac15ba	CdaC15BA								
	Yes	Auto	Running	OK	Normal	No	Yes		Kernel Driver
cdad10ba	CdaD10BA								
	Yes	Auto	Running	OK	Normal	No	Yes		Kernel Driver
cdfs	Cdfs								
	Disabled	Stopped	OK	Normal	No	No			File System Driver
cdrom	CD-ROM Driver								
	System	Stopped	OK	Normal	No	No			Kernel Driver
changer	Changer	No	Not Available	Kernel Driver	No	System	Stopped	OK	Ignore
clusdisk	Cluster Disk Driver								
	Disabled	Stopped	OK	Normal	No	No			Kernel Driver
cmdide	CmdIde	No	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
cpqcissm	cpqcissm								
	OK	Normal	No	Not Available	Kernel Driver	No	Disabled	Stopped	Kernel Driver
credisk	CRC Disk Filter Driver								
	Boot	Running	OK	Normal	No	Yes			Kernel Driver
dfsdriver	DfsDriver								
	Yes	Boot	Running	OK	Normal	No	Yes		File System Driver
disk	Disk Driver								
	Running	OK	Normal	No	Yes				Kernel Driver
dmboot	dmboot								
	Stopped	OK	Normal	No	No				Kernel Driver
dmio	Logical Disk Manager Driver								
	Yes	Boot	Running	OK	Normal	No	Yes		Kernel Driver
dmload	dmload								
	Running	OK	Normal	No	Yes				Kernel Driver
dpti2o	dpti2o	No	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
e1express	Intel(R) PRO/1000 PCI Express Network Connection Driver								
	Running	OK	Normal	No	Yes				Kernel Driver
elxstor	elxstor	No	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
em	em								
	OK	Normal	No	No					Kernel Driver
fastfat	Fastfat								
	Running	OK	Normal	No	Yes				File System Driver

	Disabled	Stopped	OK	Normal	No	No		
fdc	Fdc	e:\windows\system32\drivers\fdc.sys	Kernel Driver	No	System	Stopped		
	OK	Ignore	No	No				
fips	Fips	e:\windows\system32\drivers\fips.sys	Kernel Driver	Yes	System			
	Running	OK	Normal	No	Yes			
flpydisk	Flpydisk	e:\windows\system32\drivers\flpydisk.sys	Kernel Driver	No	System			
	Stopped	OK	Ignore	No	No			
fltmgr	FltMgr	e:\windows\system32\drivers\fltmgr.sys	File System Driver		Yes	Boot		
	Running	OK	Normal	No	Yes			
ftdisk	Volume Manager Driver	e:\windows\system32\drivers\ftdisk.sys	Kernel Driver		Yes			
	Boot	Running	OK	Normal	No	Yes		
gpc	Generic Packet Classifier	e:\windows\system32\drivers\msgpc.sys	Kernel Driver		Yes			
	Manual	Running	OK	Normal	No	Yes		
hpciss	hpciss	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
http	HTTP	e:\windows\system32\drivers\http.sys	Kernel Driver	No	Manual	Stopped		
	OK	Normal	No	No				
i2omgmt	i2omgmt	Not Available	Kernel Driver	No	System	Stopped	OK	
	Normal	No	No					
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver	e:\windows\system32\drivers\i8042prt.sys	Kernel Driver	Yes	System	Running	OK	Normal
				No	Yes			
iirsp	iirsp	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
imapi	CD-Burning Filter Driver	e:\windows\system32\drivers\imapi.sys	Kernel Driver		No			
	System	Stopped	OK	Normal	No	No		
intelide	IntelIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
intelppm	Intel Processor Driver	e:\windows\system32\drivers\intelppm.sys	Kernel Driver					
	Yes	Manual	Running	OK	Normal	No	Yes	
ip6fw	IPv6 Windows Firewall Driver	e:\windows\system32\drivers\ip6fw.sys	Kernel Driver					
	No	Manual	Stopped	OK	Normal	No	No	
ipfilterdriver	IP Traffic Filter Driver	e:\windows\system32\drivers\ipfltdrv.sys	Kernel Driver					
	No	Manual	Stopped	OK	Normal	No	No	
ipinip	IP in IP Tunnel Driver	e:\windows\system32\drivers\ipinip.sys	Kernel Driver		No			
	Manual	Stopped	OK	Normal	No	No		
ipnat	IP Network Address Translator	e:\windows\system32\drivers\ipnat.sys	Kernel Driver					
	No	Manual	Stopped	OK	Normal	No	No	
ipsec	IPSEC driver	e:\windows\system32\drivers\ipsec.sys	Kernel Driver	Yes	System			
	Running	OK	Normal	No	Yes			
isapnp	PnP ISA/EISA Bus Driver	e:\windows\system32\drivers\isapnp.sys	Kernel Driver					
	Yes	Boot	Running	OK	Critical	No	Yes	
kbdclass	Keyboard Class Driver	e:\windows\system32\drivers\kbdclass.sys	Kernel Driver					
	Yes	System	Running	OK	Normal	No	Yes	
ksecdd	KSecDD	e:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot			
	Running	OK	Normal	No	Yes			
ksthunk	Kernel Streaming WOW64 Thunk Service	e:\windows\system32\drivers\ksthunk.sys	Kernel Driver					
	Driver	Yes	Manual	Running	OK	Normal	No	Yes
lp6nds35	lp6nds35	Not Available	Kernel Driver	No	Disabled	Stopped		
	OK	Normal	No	No				
mnmdd	mnmdd	e:\windows\system32\drivers\mnmdd.sys	Kernel Driver	Yes	System			
	Running	OK	Ignore	No	Yes			
modem	Modem	e:\windows\system32\drivers\modem.sys	Kernel Driver	No	Manual	Stopped		
	OK	Ignore	No	No				

mouclass	Mouse Class Driver	e:\windows\system32\drivers\mouclass.sys	Kernel						
Driver	Yes	System Running	OK	Normal	No	Yes			
mountmgr	Mount Point Manager	e:\windows\system32\drivers\mountmgr.sys	Kernel						
Driver	Yes	Boot Running	OK	Normal	No	Yes			
mraid35x	mraid35x	Not Available	Kernel Driver	No	Disabled		Stopped		
	OK	Normal	No	No					
mrxdav	WebDav Client Redirector	e:\windows\system32\drivers\mrxdav.sys	File	System					
Driver	No	Manual Stopped	OK	Normal	No	No			
mrxsmbs	MRXSMB	e:\windows\system32\drivers\mrxsmbs.sys	File System Driver	Yes					
	System Running	OK	Normal	No	Yes				
msfs	Msfs	e:\windows\system32\drivers\msfs.sys	File System Driver	Yes	System				
	Running	OK	Normal	No	Yes				
mssmbios	Microsoft System Management BIOS	e:\windows\system32\drivers\mssmbios.sys	Kernel Driver	Yes	Manual	Driver			
	Running	OK	Normal	No	Yes				
mup	Mup	e:\windows\system32\drivers\mup.sys	File System Driver	Yes	Boot				
	Running	OK	Normal	No	Yes				
ndis	NDIS System Driver	e:\windows\system32\drivers\ndis.sys	Kernel Driver	Yes					
	Boot Running	OK	Normal	No	Yes				
ndistapi	Remote Access NDIS TAPI Driver	e:\windows\system32\drivers\ndistapi.sys	Kernel						
Driver	Yes	Manual Running	OK	Normal	No	Yes			
ndisuio	NDIS Usermode I/O Protocol	e:\windows\system32\drivers\ndisuio.sys	Kernel	Driver					
	Yes	Manual Running	OK	Normal	No	Yes			
ndiswan	Remote Access NDIS WAN Driver	e:\windows\system32\drivers\ndiswan.sys	Kernel						
Driver	Yes	Manual Running	OK	Normal	No	Yes			
ndproxy	NDIS Proxy	e:\windows\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual				
	Running	OK	Normal	No	Yes				
netbios	NetBIOS Interface	e:\windows\system32\drivers\netbios.sys	File	System	Driver				
	Yes	System Running	OK	Normal	No	Yes			
netbt	NetBios over Tcpip	e:\windows\system32\drivers\netbt.sys	Kernel Driver	Yes					
	System Running	OK	Normal	No	Yes				
nfrd960	nfrd960	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	
	No	No							
npfs	Npfs	e:\windows\system32\drivers\npfs.sys	File System Driver	Yes	System				
	Running	OK	Normal	No	Yes				
ntfs	Ntfs	e:\windows\system32\drivers\ntfs.sys	File System Driver	Yes					
	Disabled	Running	OK	Normal	No	Yes			
null	Null	e:\windows\system32\drivers\null.sys	Kernel Driver	Yes	System				
	Running	OK	Normal	No	Yes				
parport	Parport	e:\windows\system32\drivers\parport.sys	Kernel Driver	No	Manual	Stopped			
	OK	Ignore	No	No					
partmgr	Partition Manager	e:\windows\system32\drivers\partmgr.sys	Kernel Driver	Yes					
	Boot Running	OK	Normal	No	Yes				
pci	PCI Bus Driver	e:\windows\system32\drivers\pci.sys	Kernel Driver	Yes	Boot				
	Running	OK	Critical	No	Yes				
pciide	PCIIde	e:\windows\system32\drivers\pciide.sys	Kernel Driver	Yes	Boot				
	Running	OK	Normal	No	Yes				
pcmcia	Pcmcia	e:\windows\system32\drivers\pcmcia.sys	Kernel Driver	No	Disabled				
	Stopped	OK	Normal	No	No				
pdcomp	PDCOMP	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore	
	No	No							
pdframe	PDFFRAME	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore	
	No	No							

pdreli	PDRELI	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No	No						
pdrframe	PDRFRAME	Not Available	Kernel Driver	No	Manual	Stopped	OK	
	Ignore	No	No					
pmxdrv	pmxdrv	\\?\e:\windows\system32\drivers\pmxdrv.sys	Kernel Driver	No	Manual	Stopped	OK	
	Normal	No	No					
pptpminiport	WAN Miniport (PPTP)	e:\windows\system32\drivers\raspptp.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Yes	Manual	Running	OK	Normal	No	Yes	
ptilink	Direct Parallel Link Driver	e:\windows\system32\drivers\ptilink.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Yes	Manual	Running	OK	Normal	No	Yes	
ql2300	QLogic Fibre Channel	SCSI Miniport Driver	(wx64 IP)					
	e:\windows\system32\drivers\ql2300.sys	Kernel Driver	Yes	Boot	Running	OK	Normal	No
	Yes	Yes						
qldirect	qldirect	e:\windows\system32\drivers\qldirect.sys	Kernel Driver	Yes	Auto	Running	OK	Normal
	Running	OK	Normal	No	Yes			
rasacd	Remote Access Auto Connection Driver	e:\windows\system32\drivers\rasacd.sys	Kernel Driver	Yes	System	Running	OK	Normal
	Yes	System	Running	OK	Normal	No	Yes	
rasl2tp	WAN Miniport (L2TP)	e:\windows\system32\drivers\rasl2tp.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Manual	Running	OK	Normal	No	Yes		
raspppoe	Remote Access PPPOE Driver	e:\windows\system32\drivers\raspppoe.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Yes	Manual	Running	OK	Normal	No	Yes	
raspti	Direct Parallel	e:\windows\system32\drivers\raspti.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Running	OK	Normal	No	Yes			
rdbss	Rdbss	e:\windows\system32\drivers\rdbss.sys	File System Driver	Yes	System	Running	OK	Normal
	Running	OK	Normal	No	Yes			
rdpcdd	RDPcdd	e:\windows\system32\drivers\rdpcdd.sys	Kernel Driver	Yes	System	Running	OK	Ignore
	Running	OK	Ignore	No	Yes			
rdpdr	Terminal Server Device Redirector Driver	e:\windows\system32\drivers\rdpdr.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Yes	Manual	Running	OK	Normal	No	Yes	
rdpwd	RDPWD	e:\windows\system32\drivers\rdpwd.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore
	Running	OK	Ignore	No	Yes			
redbook	Digital CD Audio Playback Filter Driver	e:\windows\system32\drivers\redbook.sys	Kernel Driver	No	System	Stopped	OK	Normal
	No	System	Stopped	OK	Normal	No	No	
secdrv	Security Driver	e:\windows\system32\drivers\secdrv.sys	Kernel Driver	Yes	Auto	Running	OK	Normal
	Running	OK	Normal	No	Yes			
serenum	Serenum Filter Driver	e:\windows\system32\drivers\serenum.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Manual	Running	OK	Normal	No	Yes		
serial	Serial port driver	e:\windows\system32\drivers\serial.sys	Kernel Driver	Yes	System	Running	OK	Ignore
	Running	OK	Ignore	No	Yes			
sfloppy	Sfloppy	e:\windows\system32\drivers\sfloppy.sys	Kernel Driver	No	System	Stopped	OK	Ignore
	OK	Ignore	No	No				
simbad	Simbad	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
srv	Srv	e:\windows\system32\drivers\srv.sys	File System Driver	Yes	Manual	Running	OK	Normal
	Running	OK	Normal	No	Yes			
swenum	Software Bus Driver	e:\windows\system32\drivers\swenum.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Manual	Running	OK	Normal	No	Yes		
symc8xx	symc8xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	OK	Normal	No	No				
symmpi	symmpi	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
sym_hi	sym_hi	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						

sym_u3	sym_u3	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
tcpip	TCP/IP Protocol Driver	System Running	e:\windows\system32\drivers\tcpip.sys	Kernel Driver	Yes			Yes
tdpipe	TDPIPE	Ignore	e:\windows\system32\drivers\tdpipe.sys	Kernel Driver	No	Manual	Stopped	
	OK	No						
tdtcp	TDTCP	Running	e:\windows\system32\drivers\tdtcp.sys	Kernel Driver	Yes	Manual		
	OK	Ignore			No	Yes		
termdd	Terminal Device Driver	System Running	e:\windows\system32\drivers\termdd.sys	Kernel Driver	Yes			Yes
	OK	Normal			No	Yes		
toside	TosIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
udfs	Udfs	Disabled	e:\windows\system32\drivers\udfs.sys	File System Driver	No			
	Stopped	OK			Normal	No		
ultra	ultra	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
update	Microcode Update Driver	Yes	e:\windows\system32\drivers\update.sys	Kernel Driver	Yes			
	Manual	Running			OK	Normal	No	Yes
usbhci	Microsoft USB 2.0 Enhanced Host Controller	Miniport Driver	e:\windows\system32\drivers\usbhci.sys	Kernel Driver	Yes	Manual	Running	
	OK	Normal			No	Yes		
usbhub	USB2 Enabled Hub	Manual	e:\windows\system32\drivers\usbhub.sys	Kernel Driver	Yes			Yes
	Running	OK			Normal	No	Yes	
usbuhci	Microsoft USB Universal Host Controller	Miniport Driver	e:\windows\system32\drivers\usbuhci.sys	Kernel Driver	Yes	Manual	Running	
	OK	Normal			No	Yes		
vga	vga	Running	e:\windows\system32\drivers\vgapnp.sys	Kernel Driver	Yes	Manual		
	OK	Ignore			No	Yes		
vgasave	VGA Display Controller.	System Stopped	e:\windows\system32\drivers\vga.sys	Kernel Driver	No			
	OK	Ignore			No	No		
viaide	ViaIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
volsnap	Storage volumes	Running	e:\windows\system32\drivers\volsnap.sys	Kernel Driver	Yes	Boot		
	OK	Normal			No	Yes		
wanarp	Remote Access IP ARP Driver	Yes	e:\windows\system32\drivers\wanarp.sys	Kernel Driver	Yes			
	Manual	Running			OK	Normal	No	Yes
wdica	WDICANot Available	No	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No							No
wlbs	Network Load Balancing	Manual	e:\windows\system32\drivers\wlbs.sys	Kernel Driver	No			
	Stopped	OK			Normal	No	No	

[Signed Drivers]

Device Name	Signed	Device Class	Driver Version	Driver Date	Manufacturer	INF
Name	Driver Name	Device ID				
QLogic	Optimizing and Multipath Driver	No	SCSIADAPTER	Not Available	Not Available	Available
	Not Available	oem3.inf	Not Available	ROOT\UNKNOWN\0000		
Microsoft	System Management BIOS Driver	Yes	SYSTEM	5.2.3790.1830		
	10/1/2002	(Standard system devices)	machine.inf	Not		Available
	ROOT\SYSTEM\0002					
Microcode Update Device	Yes	SYSTEM	5.2.3790.1830	10/1/2002		
	(Standard system devices)	machine.inf	Not Available	ROOT\SYSTEM\0001		
Plug and Play Software Device Enumerator	Yes	SYSTEM	5.2.3790.1830			

	10/1/2002	(Standard system devices)	machine.inf	Not	Available
	ROOT\SYSTEM\0000				
Terminal Server Mouse Driver	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not Available	ROOT\RDP_MOU\0000	
Terminal Server Keyboard Driver	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not Available	ROOT\RDP_KBD\0000	
Terminal Server Device Redirector	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not Available	ROOT\RDPDR\0000	
Direct Parallel	Yes	NET	5.2.3790.1830	10/1/2002	Microsoft
Not Available		ROOT\MS_PTMINIPORT\0000			netrasa.inf
WAN Miniport (PPTP)	Yes	NET	5.2.3790.1830	10/1/2002	Microsoft
netrasa.inf		Not Available	ROOT\MS_PPTPMINIPORT\0000		
WAN Miniport (PPPOE)	Yes	NET	5.2.3790.1830	10/1/2002	Microsoft
netrasa.inf		Not Available	ROOT\MS_PPPOEMINIPORT\0000		
WAN Miniport (IP)	Yes	NET	5.2.3790.1830	10/1/2002	Microsoft
netrasa.inf		Not Available	ROOT\MS_NDISWANIP\0000		
WAN Miniport (L2TP)	Yes	NET	5.2.3790.1830	10/1/2002	Microsoft
netrasa.inf		Not Available	ROOT\MS_L2TPMINIPORT\0000		
Video Codecs	Yes	MEDIA	5.2.3790.1830	10/1/2002	(Standard system devices)
wave.inf		Not Available	ROOT\MEDIA\MS_MMVID		
Legacy Video Capture Devices	Yes	MEDIA	5.2.3790.1830	10/1/2002	(Standard system devices)
wave.inf		Not Available	ROOT\MEDIA\MS_MMVCD		
Media Control Devices	Yes	MEDIA	5.2.3790.1830	10/1/2002	(Standard system devices)
wave.inf		Not Available	ROOT\MEDIA\MS_MMMCI		
Legacy Audio Drivers	Yes	MEDIA	5.2.3790.1830	10/1/2002	(Standard system devices)
wave.inf		Not Available	ROOT\MEDIA\MS_MMDRV		
Audio Codecs	Yes	MEDIA	5.2.3790.1830	10/1/2002	(Standard system devices)
wave.inf		Not Available	ROOT\MEDIA\MS_MMACM		
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER		Not Available	Not Available
Available	Not Available	Not Available	Not		Available
	ROOT\LEGACY_WANARP\0000				
volsnap	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_VOLSNAP\0000		Available
TDTCP	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_TDTCP\0000		Available
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_TCPIP\0000		Available
Security Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Available		Not Available	ROOT\LEGACY_SECDRV\0000		Not Available
RDPWD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Available		Not Available	ROOT\LEGACY_RDPWD\0000		Not Available
RDPCDD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Available		Not Available	ROOT\LEGACY_RDPCDD\0000		Not Available
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	Not Available	Not	Available
	ROOT\LEGACY_RASACD\0000				
pmxdrv	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_PMXDRV\0000		Available
Partition Manager	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_PARTMGR\0000		Available
Null	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available
Not Available		Not Available	ROOT\LEGACY_NULL\0000		Available
NetBios over Tcpi	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available

	Not Available	Not Available	Not Available	ROOT\LEGACY_NETBT\0000		
NDProxy	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOT\LEGACY_NDPROXY\0000			
NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	Not		Available	
			ROOT\LEGACY_NDISUIO\0000			
Remote Access NDIS TAPI Driver		Not Available	LEGACYDRIVER	Not	Available	
	Not Available	Not Available	Not Available	Not	Available	
			ROOT\LEGACY_NDISTAPI\0000			
NDIS System Driver	Not Available	LEGACYDRIVER	Not Available	Not	Available	
	Not Available	Not Available	Not Available	ROOT\LEGACY_NDIS\0000		
mountmgr	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOT\LEGACY_MOUNTMGR\0000			
mmdd	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_MNMDD\0000			
ksecdd	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_KSECDD\0000			
IPSEC driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOT\LEGACY_IPSEC\0000			
IP Network Address Translator	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	Not Available	ROOT\LEGACY_IPNAT\0000		
Generic Packet Classifier	Not Available	LEGACYDRIVER	Not Available	Not	Available	
	Not Available	Not Available	Not Available	ROOT\LEGACY_GPC\0000		
Fips	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_FIPS\0000			
em	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_EM\0000			
dmload	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_DMLOAD\0000			
dmboot	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_DMBOOT\0000			
CRC Disk Filter Driver	Not Available	LEGACYDRIVER	Not Available	Not	Available	
	Not Available	Not Available	Not Available	ROOT\LEGACY_CRCDISK\0000		
CdaD10BA	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOT\LEGACY_CDAD10BA\0000			
CdaC15BA	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOT\LEGACY_CDAC15BA\0000			
Beep	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_BEEP\0000			
AFD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOT\LEGACY_AFD\0000			
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf		Not				Available
			STORAGE\VOLUME\1&30A96598&0&SIGNATURE1OFFSETEA58C3E00LENGTHD4A			
			6FBA00			
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf		Not				Available
			STORAGE\VOLUME\1&30A96598&0&SIGNATURE1OFFSET752C65E00LENGTH752C			
			56200			
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf		Not				Available
			STORAGE\VOLUME\1&30A96598&0&SIGNATURE1OFFSET7E00LENGTH752C56200			
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	

volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1632OFFSET520EDEC00LEN						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1632OFFSET3EC10000LENGT						
H4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1632OFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16C4OFFSET520EDEC00LEN						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16C4OFFSET3EC10000LENG						
TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16C4OFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET520EDEC00LEN						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET3EC10000LENG						
TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET520EDEC00LEN						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET3EC10000LENG						
TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16D0OFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F8OFFSET520EDEC00LEN						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available

STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F8OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F8OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16BCOFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16BCOFFSET3EC10000LENG
TH4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16BCOFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1748OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1748OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1748OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1656OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1656OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1656OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1658OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1658OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1658OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1662OFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1662OFFSET3EC10000LENGT

H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1662OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1674OFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1674OFFSET3EC10000LENGT

H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1674OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD167E0FFSET7D0C56200LEN

GTH6ED41E9E00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD167E0FFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD167E0FFSET3EC10000LENG

TH4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD167E0FFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1600OFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1600OFFSET3EC10000LENGT

H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1600OFFSET7E00LENGTH3E

C08200

Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD160AOFFSET7D0C56200LENG						
GTH6ED41E9E00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD160AOFFSET520EDEC00LENG						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD160AOFFSET3EC10000LENG						
TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD160AOFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD161COFFSET520EDEC00LENG						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD161COFFSET3EC10000LENG						
TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD161COFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1626OFFSET7D0C56200LENG						
TH6ED41E9E00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1626OFFSET520EDEC00LENG						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1626OFFSET3EC10000LENGT						
H4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1626OFFSET7E00LENGTH3E						
C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1628OFFSET520EDEC00LENG						
GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1628OFFSET3EC10000LENGT						
H4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	

	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1628OFFSET7E00LENGTH3E					
C08200						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1694OFFSET7D0C56200LENG					
TH6ED41E9E00						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1694OFFSET520EDEC00LEN					
GTH2AFD77600						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1694OFFSET3EC10000LENGT					
H4E22CEC00						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1694OFFSET7E00LENGTH3E					
C08200						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD169EOFFSET520EDEC00LEN					
GTH2AFD77600						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD169EOFFSET3EC10000LENG					
TH4E22CEC00						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD169EOFFSET7E00LENGTH3E					
C08200						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATUREA411377EOFFSET7E00LENGTH76					
A4E38200						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16CEOFFSET520EDEC00LEN					
GTH2AFD77600						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16CEOFFSET3EC10000LENG					
TH4E22CEC00						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16CEOFFSET7E00LENGTH3E					
C08200						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available
	STORAGE\VOLUME\1&30A96598&0&SIGNATURE79E6758EOFFSET520EDEC00LEN					
GTH2AFD77600						
Generic	volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
	volume.inf	Not				Available

STORAGE\VOLUME\1&30A96598&0&SIGNATURE79E6758EOFFSET3EC10000LENG
TH4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE79E6758EOFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F6OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F6OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16F6OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1682OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1682OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1682OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1746OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1746OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1746OFFSET7E00LENGTH3E
C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1752OFFSET520EDEC00LEN
GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1752OFFSET3EC10000LENGT
H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1752OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16A0OFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16A0OFFSET3EC10000LENG

TH4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16A0OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD176EOFFSET7D0C56200LEN

GTH6ED41E9E00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD176EOFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD176EOFFSET3EC10000LENG

TH4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD176EOFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1770OFFSET7D0C56200LENG

TH6ED41E9E00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1770OFFSET520EDEC00LEN

GTH2AFD77600
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1770OFFSET3EC10000LENGT

H4E22CEC00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1770OFFSET7E00LENGTH3E

C08200
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD177AOFFSET7D0C56200LEN

GTH6ED41E9E00
Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD177AOFFSET520EDEC00LEN

GTH2AFD77600

Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD177AOFFSET3EC10000LENG TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD177AOFFSET7E00LENGTH3E C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD170COFFSET7D0C56200LEN GTH6ED41E9E00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD170COFFSET520EDEC00LEN GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD170COFFSET3EC10000LENG TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD170COFFSET7E00LENGTH3E C08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B4OFFSET7D0C56200LEN GTH6ED41E9E00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B4OFFSET520EDEC00LE NGTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B4OFFSET3EC10000LENG TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B4OFFSET7E00LENGTH3 EC08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16AAOFFSET520EDEC00LEN GTH2AFD77600						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16AAOFFSET3EC10000LENG TH4E22CEC00						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	
volume.inf	Not					Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD16AAOFFSET7E00LENGTH3 EC08200						
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft	

volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1A3OFFSET7D0C56200LEN					
GTH6ED41E9E00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1A3OFFSET520EDEC00LE					
NGTH2AFD77600					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1A3OFFSET3EC10000LEN					
GTH4E22CEC00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1A3OFFSET7E00LENGTH3					
EC08200					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B18AOFFSET7D0C56200LEN					
GTH6ED41E9E00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B18AOFFSET520EDEC00LE					
NGTH2AFD77600					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B18AOFFSET3EC10000LEN					
GTH4E22CEC00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B18AOFFSET7E00LENGTH3					
EC08200					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B9OFFSET7D0C56200LEN					
GTH6ED41E9E00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B9OFFSET520EDEC00LE					
NGTH2AFD77600					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B9OFFSET3EC10000LENG					
TH4E22CEC00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATUREEC00B1B9OFFSET7E00LENGTH3					
EC08200					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available
STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1716OFFSET7D0C56200LENG					
TH6ED41E9E00					
Generic volume	Yes	VOLUME	5.2.3790.1830	10/1/2002	Microsoft
volume.inf	Not				Available

STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1716OFFSET520EDEC00LEN
GTH2AFD77600

Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available

STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1716OFFSET3EC10000LENGT
H4E22CEC00

Generic volume Yes VOLUME 5.2.3790.1830 10/1/2002 Microsoft
volume.inf Not Available

STORAGE\VOLUME\1&30A96598&0&SIGNATURE9AD1716OFFSET7E00LENGTH3E
C08200

Volume Manager Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available ROOT\FTDISK\0000

Logical Disk Manager Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available ROOT\DMIO\0000

ACPI Fixed Feature Button Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available

ACPI\FIXEDBUTTON\2&DABA3FF&0

Advanced programmable interrupt controller Yes SYSTEM 5.2.3790.1830
10/1/2002 (Standard system devices) machine.inf Not Available

ACPI\PNP0003\3&61AAA01&0

Motherboard resources Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\PNP0C02\3&61AAA01&0

Intel(R) 631xESB/6321ESB SMBus Controller - 269B Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_269B&SUBSYS_34788086&REV_09\3&61AAA01&0&FB

Secondary IDE Channel Yes HDC 5.2.3790.1830 10/1/2002 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available

PCIIDE\IDECHANNEL\4&6BD4A7B&0&1

Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available

IDE\DISKMAXTOR_6Y120M0_____YAR51EW0\3359414C5458
454A202020202020202020202020

Primary IDE Channel Yes HDC 5.2.3790.1830 10/1/2002 (Standard IDE
ATA/ATAPI controllers) mshdc.inf Not Available

PCIIDE\IDECHANNEL\4&6BD4A7B&0&0

Intel(R) 631xESB/6321ESB Serial ATA Storage Controller - 2680 Yes HDC 7.3.0.1010
11/18/2005 Intel esb2ide.inf Not Available

PCI\VEN_8086&DEV_2680&SUBSYS_34788086&REV_09\3&61AAA01&0&FA

High precision event timer Yes SYSTEM 5.2.3790.1830 10/1/2002
(Standard system devices) machine.inf Not Available

ACPI\PNP0103\4&2AA4AD3D&0

Communications Port Yes PORTS 5.2.3790.1830 10/1/2002 (Standard port types)
msports.inf Not Available ACPI\PNP0501\2

Communications Port Yes PORTS 5.2.3790.1830 10/1/2002 (Standard port types)
msports.inf Not Available ACPI\PNP0501\1

Standard 101/102-Key or Microsoft Natural PS/2 Keyboard Yes KEYBOARD 5.2.3790.1830
10/1/2002 (Standard keyboards) keyboard.inf Not Available

ACPI\PNP0303\4&2AA4AD3D&0

Microsoft PS/2 Mouse Yes MOUSE 5.2.3790.1830 10/1/2002 Microsoft
msmouse.inf Not Available ACPI\PNP0F03\4&2AA4AD3D&0

Motherboard resources Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\PNP0C02\4&2AA4AD3D&0

System speaker Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)

machine.inf	Not Available	ACPI\PNP0800\4&2AA4AD3D&0			
System timer	Yes	SYSTEM	5.2.3790.1830	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\PNP0100\4&2AA4AD3D&0			
Numeric data processor	Yes	SYSTEM	5.2.3790.1830	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\PNP0C04\4&2AA4AD3D&0			
Programmable interrupt controller	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
		ACPI\PNP0000\4&2AA4AD3D&0			
System CMOS/real time clock	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
		ACPI\PNP0B00\4&2AA4AD3D&0			
Direct memory access controller	Yes	SYSTEM	5.2.3790.1830	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
		ACPI\PNP0200\4&2AA4AD3D&0			
Intel(R) 631xESB/6321ESB LPC Interface Controller - 2670	Yes	SYSTEM			
7.3.0.1010	11/18/2005	Intel	5000xzvp.inf	Not	Available
		PCI\VEN_8086&DEV_2670&SUBSYS_00000000&REV_09\3&61AAA01&0&F8			
Default Monitor	Yes	MONITOR	5.2.3790.1830	10/1/2002	(Standard monitor types)
monitor.inf	Not				Available
		DISPLAY\DEFAULT_MONITOR\5&27F04233&0&12345678&0D&0C			
Standard VGA Graphics Adapter	Yes	DISPLAY	5.2.3790.1830	10/1/2002	
(Standard display types)		display.inf	Not		Available
		PCI\VEN_1002&DEV_515E&SUBSYS_34788086&REV_02\4&2014205D&0&60F0			
Intel(R) 82801 PCI Bridge - 244E	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel
machine.inf	Not				Available
		PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_D9\3&61AAA01&0&F0			
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002	(Standard USB Host Controller)
usbport.inf	Not Available	USB\ROOT_HUB20\4&11F99B2B&0			
Intel(R) 631xESB/6321ESB USB2 Enhanced Host Controller - 268C	Yes	USB			
7.3.0.1010	11/18/2005	Intel	esb2usb.inf	Not	Available
		PCI\VEN_8086&DEV_268C&SUBSYS_34788086&REV_09\3&61AAA01&0&EF			
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002	(Standard USB Host Controller)
usbport.inf	Not Available	USB\ROOT_HUB\4&3032C334&0			
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268B	Yes	USB			
7.3.0.1010	11/18/2005	Intel	esb2usb.inf	Not	Available
		PCI\VEN_8086&DEV_268B&SUBSYS_34788086&REV_09\3&61AAA01&0&EB			
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002	(Standard USB Host Controller)
usbport.inf	Not Available	USB\ROOT_HUB\4&19D90277&0			
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 268A	Yes	USB			
7.3.0.1010	11/18/2005	Intel	esb2usb.inf	Not	Available
		PCI\VEN_8086&DEV_268A&SUBSYS_34788086&REV_09\3&61AAA01&0&EA			
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002	(Standard USB Host Controller)
usbport.inf	Not Available	USB\ROOT_HUB\4&62221E4&0			
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2689	Yes	USB			
7.3.0.1010	11/18/2005	Intel	esb2usb.inf	Not	Available
		PCI\VEN_8086&DEV_2689&SUBSYS_34788086&REV_09\3&61AAA01&0&E9			
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002	(Standard USB Host Controller)
usbport.inf	Not Available	USB\ROOT_HUB\4&84ED9C0&0			
Intel(R) 631xESB/6321ESB USB Universal Host Controller - 2688	Yes	USB			
7.3.0.1010	11/18/2005	Intel	esb2usb.inf	Not	Available
		PCI\VEN_8086&DEV_2688&SUBSYS_34788086&REV_09\3&61AAA01&0&E8			
Qlogic processor device	Yes	SYSTEM	5.2.3790.1830	10/1/2002	QLOGIC
csasidev.inf	Not				Available

SCSIPROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5&161024FD&0
&07F0

Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&060					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&050					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&040					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&030					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&020					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&010					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&161024FD&0&000					
QLogic Fibre Channel Adapter	No	SCSIADAPTER	9.1.0.12	9/21/2005	QLogic
oem1.inf	Not				Available
PCI&VEN_1077&DEV_2432&SUBSYS_01381077&REV_024&187919FE&0&01E0					
Qlogic processor device	Yes	SYSTEM	5.2.3790.1830	10/1/2002	QLOGIC
scsidev.inf	Not				Available
SCSIPROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_5&CF17D60&0 &07F0					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&0B0					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&0A0					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&090					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&080					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&070					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&060					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available
SCSIDISK&VEN_EUROLOGC&PROD_FC2502&REV_79025&CF17D60&0&050					
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002	(Standard disk drives)
disk.inf	Not				Available

SCSISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\5&CF17D60&0&040
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available

SCSISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\5&CF17D60&0&030
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available

SCSISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\5&CF17D60&0&020
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available

SCSISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\5&CF17D60&0&010
Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
disk.inf Not Available

SCSISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\5&CF17D60&0&000
QLogic Fibre Channel Adapter No SCSIADAPTER 9.1.0.12 9/21/2005 QLogic
oem1.inf Not Available

PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\4&187919FE&0&00E0
Intel(R) 631xESB/6321ESB PCI Express Root Port 1 - 2690 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_2690&SUBSYS_00000000&REV_09\3&61AAA01&0&E0
Intel(R) 5000 Series Chipset FBD Registers - 25F6 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F6&SUBSYS_00000000&REV_92\3&61AAA01&0&B0
Intel(R) 5000 Series Chipset FBD Registers - 25F5 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F5&SUBSYS_00000000&REV_92\3&61AAA01&0&A8
Intel(R) 5000 Series Chipset Reserved Registers - 25F3 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F3&SUBSYS_00000000&REV_92\3&61AAA01&0&98
Intel(R) 5000 Series Chipset Reserved Registers - 25F1 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F1&SUBSYS_00000000&REV_92\3&61AAA01&0&88
Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_92\3&61AAA01&0&82
Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_92\3&61AAA01&0&81
Intel(R) 5000 Series Chipset Error Reporting Registers - 25F0 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F0&SUBSYS_00000000&REV_92\3&61AAA01&0&80
Base System Device Not Available UNKNOWN Not Available Not Available Not Available
Available Not Available Not Available Available

PCI\VEN_8086&DEV_1A38&SUBSYS_34788086&REV_92\3&61AAA01&0&40
Intel(R) 5000 Series Chipset PCI Express x4 Port 7 - 25E7 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25E7&SUBSYS_00000000&REV_92\3&61AAA01&0&38
Intel(R) 5000 Series Chipset PCI Express x8 Port 6-7 - 25F9 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25F9&SUBSYS_00000000&REV_92\3&61AAA01&0&30
Intel(R) 5000 Series Chipset PCI Express x4 Port 5 - 25E5 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available

PCI\VEN_8086&DEV_25E5&SUBSYS_00000000&REV_92\3&61AAA01&0&28
Intel(R) 5000 Series Chipset PCI Express x4 Port 4 - 25E4 Yes SYSTEM 7.3.0.1010

	11/18/2005	Intel	5000xzvp.inf	Not			Available
			PCI\VEN_8086&DEV_25E4&SUBSYS_00000000&REV_92\3&61AAA01&0&20				
Intel(R) 5000 Series Chipset PCI Express x4 Port 3 - 25E3	Yes	SYSTEM	7.3.0.1010				
	11/18/2005	Intel	5000xzvp.inf	Not			Available
			PCI\VEN_8086&DEV_25E3&SUBSYS_00000000&REV_92\3&61AAA01&0&18				
Intel(R) 6311ESB/6321ESB PCI Express to PCI-X Bridge - 350C	Yes	SYSTEM					
	7.3.0.1010	11/18/2005	Intel	5000xzvp.inf	Not		Available
			PCI\VEN_8086&DEV_350C&SUBSYS_00000000&REV_01\4&25C5D7DF&0&0310				
PCI Device	Not Available	UNKNOWN	Not Available	Not Available	Not Available	Not Available	Available
	Not Available	Not					Available
			PCI\VEN_8086&DEV_1089&SUBSYS_00008086&REV_01\6&AA0C0A6&0&07100010				
USB Root Hub	Yes	USB	5.2.3790.1830	10/1/2002		(Standard USB Host Controller)	
	usbport.inf	Not Available	USB\ROOT_HUB\7&19266F24&0				
Standard Universal PCI to USB Host Controller	Yes	USB	5.2.3790.1830	10/1/2002			
	(Standard USB Host Controller)	usbport.inf	Not				Available
			PCI\VEN_8086&DEV_1087&SUBSYS_00008086&REV_01\6&AA0C0A6&0&05100010				
PCI Device	Not Available	UNKNOWN	Not Available	Not Available	Not Available	Not Available	Available
	Not Available	Not					Available
			PCI\VEN_8086&DEV_1086&SUBSYS_00008086&REV_01\6&AA0C0A6&0&04100010				
PCI Serial Port	Not Available	UNKNOWN	Not Available	Not Available	Not Available	Not Available	Available
	Not Available	Not					Available
			PCI\VEN_8086&DEV_1085&SUBSYS_00008086&REV_01\6&AA0C0A6&0&03100010				
Secondary IDE Channel	Yes	HDC	5.2.3790.1830	10/1/2002		(Standard IDE	
ATA/ATAPI controllers)	mshdc.inf	Not					Available
			PCI\IDE\IDECHANNEL\7&D715D63&0&1				
Primary IDE Channel	Yes	HDC	5.2.3790.1830	10/1/2002		(Standard IDE	
ATA/ATAPI controllers)	mshdc.inf	Not					Available
			PCI\IDE\IDECHANNEL\7&D715D63&0&0				
Standard Dual Channel PCI IDE Controller	Yes	HDC	5.2.3790.1830	10/1/2002			
	(Standard IDE ATA/ATAPI controllers)	mshdc.inf	Not				Available
			PCI\VEN_8086&DEV_1084&SUBSYS_00008086&REV_01\6&AA0C0A6&0&02100010				
Intel(R) PRO/1000 EB Network Connection with I/O Acceleration	No	NET	9.3.28.0				
	1/23/2006	Intel	oem0.inf	Not			Available
			PCI\VEN_8086&DEV_1096&SUBSYS_34768086&REV_01\6&AA0C0A6&0&01100010				
Intel(R) PRO/1000 EB Network Connection with I/O Acceleration	No	NET	9.3.28.0				
	1/23/2006	Intel	oem0.inf	Not			Available
			PCI\VEN_8086&DEV_1096&SUBSYS_34768086&REV_01\6&AA0C0A6&0&00100010				
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E3 - 3518	Yes	SYSTEM					
	7.3.0.1010	11/18/2005	Intel	5000xzvp.inf	Not		Available
			PCI\VEN_8086&DEV_3518&SUBSYS_00000000&REV_01\5&CC6F5AA&0&100010				
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E2 - 3514	Yes	SYSTEM					
	7.3.0.1010	11/18/2005	Intel	5000xzvp.inf	Not		Available
			PCI\VEN_8086&DEV_3514&SUBSYS_00000000&REV_01\5&CC6F5AA&0&080010				
Qlogic processor device	Yes	SYSTEM	5.2.3790.1830	10/1/2002		QLOGIC	
	scsudev.inf	Not					Available
			SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\7&1C548EE3&0&07F0				
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002		(Standard disk drives)	
	disk.inf	Not					Available
			SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&060				
Disk drive	Yes	DISKDRIVE	5.2.3790.1830	10/1/2002		(Standard disk drives)	
	disk.inf	Not					Available
			SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&050				

Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&040
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&030
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&020
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&010
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&1C548EE3&0&000
 QLogic Fibre Channel Adapter No SCSIADAPTER 9.1.0.12 9/21/2005 QLogic
 oem1.inf Not Available
 PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\6&AECCA84&0&01000010
 Qlogic processor device Yes SYSTEM 5.2.3790.1830 10/1/2002 QLOGIC
 scsidev.inf Not Available
 SCSI\PROCESSOR&VEN_QLOGIC&PROD_PSEUDO_DEVICE&REV_\7&70C6A77&0
 &07F0
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&0A0
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&090
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&080
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&070
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&060
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&050
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&040
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&030
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&020
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)
 disk.inf Not Available
 SCSI\DISK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&010
 Disk drive Yes DISKDRIVE 5.2.3790.1830 10/1/2002 (Standard disk drives)

```

disk.inf Not Available
SCSISDK&VEN_EUROLOGC&PROD_FC2502&REV_7902\7&70C6A77&0&000
QLogic Fibre Channel Adapter No SCSIAADAPTER9.1.0.12 9/21/2005 QLogic
oem1.inf Not Available
PCI\VEN_1077&DEV_2432&SUBSYS_01381077&REV_02\6&AECCA84&0&00000010
Intel(R) 6311ESB/6321ESB PCI Express Downstream Port E1 - 3510 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available
PCI\VEN_8086&DEV_3510&SUBSYS_00000000&REV_01\5&CC6F5AA&0&000010
Intel(R) 6311ESB/6321ESB PCI Express Upstream Port - 3500 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available
PCI\VEN_8086&DEV_3500&SUBSYS_00000000&REV_01\4&25C5D7DF&0&0010
Intel(R) 5000 Series Chipset PCI Express x8 Port 2-3 - 25F7 Yes SYSTEM
7.3.0.1010 11/18/2005 Intel 5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25F7&SUBSYS_00000000&REV_92\3&61AAA01&0&10
Intel(R) 5000P Chipset Memory Controller Hub - 25D8 Yes SYSTEM 7.3.0.1010
11/18/2005 Intel 5000xzvp.inf Not Available
PCI\VEN_8086&DEV_25D8&SUBSYS_00000000&REV_92\3&61AAA01&0&00
PCI bus Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard system devices)
machine.inf Not Available ACPI\PNP0A03\2&DABA3FF&0
ACPI Sleep Button Yes SYSTEM 5.2.3790.1830 10/1/2002 (Standard
system devices) machine.inf Not Available ACPI\PNP0C0E\2&DABA3FF&0
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_7
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_6
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_5
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_4
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_3
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_2
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_1
Intel Processor Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
Available ACPI\GENUINEINTEL_-EM64T_FAMILY_15_MODEL_6_0
Microsoft ACPI-Compliant System Yes SYSTEM 5.2.3790.1830 10/1/2002
Microsoft acpi.inf Not Available ACPI_HAL\PNP0C08\0
ACPI Multiprocessor x64-based PC Yes COMPUTER 5.2.3790.1830 10/1/2002
(Standard computers) hal.inf Not Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not Available Not Available Not Available
Not Available Not Available HTREE\ROOT\0

```

[Environment Variables]

```

Variable Value User Name
ClusterLog E:\WINDOWS\Cluster\cluster.log<SYSTEM>
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
lib E:\Program Files\SQLXML 4.0\bin\ <SYSTEM>
NUMBER_OF_PROCESSORS 8 <SYSTEM>
OS Windows_NT <SYSTEM>

```

Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;E:\Program Files (x86)\Microsoft SQL Server\80\Tools\Binn\;E:\Program Files\Microsoft SQL Server\90\DTS\Binn\;E:\Program Files\Microsoft SQL Server\90\Tools\bin\;E:\Program Files (x86)\Microsoft SQL Server\90\DTS\Binn\;E:\Program Files (x86)\Microsoft SQL Server\90\Tools\Binn\VSShell\Common7\IDE\;E:\Program Files (x86)\Microsoft Visual Studio 8\Common7\IDE\PrivateAssemblies\;E:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn <SYSTEM>

PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>

PROCESSOR_ARCHITECTURE AMD64<SYSTEM>

PROCESSOR_IDENTIFIER EM64T Family 15 Model 6 Stepping 4, GenuineIntel <SYSTEM>

PROCESSOR_LEVEL 15 <SYSTEM>

PROCESSOR_REVISION 0604 <SYSTEM>

TEMP %SystemRoot%\TEMP <SYSTEM>

TMP %SystemRoot%\TEMP <SYSTEM>

windir %SystemRoot% <SYSTEM>

TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM

TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM

TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE

TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE

TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE

TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE

TEMP %USERPROFILE%\Local Settings\Temp TPCCX64\Administrator

TMP %USERPROFILE%\Local Settings\Temp TPCCX64\Administrator

[Print Jobs]

Document	Size	Owner	Notify	Status	Time Submitted	Start Time	Until	Time
	Elapsed Time	Pages Printed	Job ID	Priority	Parameters	Driver	Print	Processor
	Host Print Queue		Data Type	Name				

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
B:	\\192.168.0.161\mykit	Disk	Persistent Connection	TPCCX64\Administrator

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start
Time	Version	Size	File Date			
system idle process		Not Available	0	0	Not Available	Not Available
Available		Not Available	Not Available	Not Available		Not
system	Not Available	4	8	0	1413120	Not Available
	Not Available	Not Available				Not Available
smss.exe	Not Available	996	11	204800	1413120	4/3/2006 12:57 PM
	Not Available	Not Available	Not Available			
csrss.exe	Not Available	412	13	Not Available	Not Available	4/3/2006 12:57 PM
	Not Available	Not Available	Not Available			
winlogon.exe	e:\windows\system32\winlogon.exe		560	13	204800	1413120
	4/3/2006 12:57 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)			901.00	KB
(922,624 bytes)	3/25/2005 8:00 PM					
services.exe	e:\windows\system32\services.exe		688	9	204800	1413120

	4/3/2006 12:57 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	216.50	KB
(221,696 bytes)	3/25/2005 8:00 PM			
lsass.exe	e:\windows\system32\lsass.exe	716 9 204800 1413120		
	4/3/2006 12:57 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB
(14,336 bytes)	3/25/2005 8:00 PM			
svchost.exe	e:\windows\system32\svchost.exe	208 8 204800 1413120		
	4/3/2006 12:57 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.50	KB
(25,088 bytes)	3/25/2005 8:00 PM			
svchost.exe	Not Available	440 8 Not Available Not Available	4/3/2006	12:58 PM
svchost.exe	Not Available	608 8 Not Available Not Available	4/3/2006	12:58 PM
svchost.exe	Not Available	660 8 Not Available Not Available	4/3/2006	12:58 PM
svchost.exe	e:\windows\system32\svchost.exe	724 8 204800 1413120		
	4/3/2006 12:58 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.50	KB
(25,088 bytes)	3/25/2005 8:00 PM			
spoolsv.exe	e:\windows\system32\spoolsv.exe	1332 8 204800 1413120		
	4/3/2006 12:58 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	107.00	KB
(109,568 bytes)	3/25/2005 8:00 PM			
msdtc.exe	Not Available	1364 8 Not Available Not Available	4/3/2006	12:58 PM
svchost.exe	e:\windows\system32\svchost.exe	1584 8 204800 1413120		
	4/3/2006 12:58 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.50	KB
(25,088 bytes)	3/25/2005 8:00 PM			
msdtssrvr.exe	Not Available	1608 8 Not Available Not Available	4/3/2006	12:58 PM
svchost.exe	Not Available	988 8 Not Available Not Available	4/3/2006	12:58 PM
msftesql.exe	e:\program files\microsoft sql server\mssql.1\mssql\binn\msftesql.exe	1272		
	8 204800 1413120	4/3/2006 12:58 PM	12.0.5626.1	152.20 KB
(155,856 bytes)	8/26/2005 5:17 PM			
svchost.exe	e:\windows\system32\svchost.exe	1628 8 204800 1413120		
	4/3/2006 12:58 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.50	KB
(25,088 bytes)	3/25/2005 8:00 PM			
wmiprvse.exe	Not Available	1384 8 Not Available Not Available	4/3/2006	12:59 PM
csrss.exe	Not Available	2032 13 Not Available Not Available	4/3/2006	1:01 PM
winlogon.exe	e:\windows\system32\winlogon.exe	1136 13 204800 1413120		
	4/3/2006 1:01 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	901.00	KB
(922,624 bytes)	3/25/2005 8:00 PM			
rdpclip.exe	e:\windows\system32\rdpclip.exe	1800 8 204800 1413120		
	4/3/2006 1:02 PM	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	99.00	KB
(101,376 bytes)	3/30/2006 10:23 AM			
explorer.exe	e:\windows\explorer.exe	1184 8 204800 1413120	4/3/2006	1:02 PM
	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.30 MB	(1,364,480 bytes)	
	3/25/2005 8:00 PM			
cmd.exe	e:\windows\system32\cmd.exe	2252 8 204800 1413120	4/3/2006	1:05 PM
	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	538.50 KB	(551,424 bytes)	
	3/25/2005 8:00 PM			
logon.scr	Not Available	1864 4 Not Available Not Available	4/3/2006	1:08 PM
	Not Available	Not Available		

```

helpctr.exe      e:\windows\pchealth\helpctr\binaries\helpctr.exe 2548 8 204800
                1413120 4/4/2006 9:10 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
                1.30 MB (1,363,456 bytes) 3/30/2006 10:25 AM
wmiprvse.exe    Not Available 2124 8 Not Available Not Available 4/4/2006 9:10
AM Not Available Not Available Not Available
helpsvc.exe     e:\windows\pchealth\helpctr\binaries\helpsvc.exe 2160 8 204800
                1413120 4/4/2006 9:10 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
                1.52 MB (1,591,296 bytes) 3/30/2006 10:25 AM
mmc.exe         e:\windows\system32\mmc.exe 592 8 204800 1413120
                4/4/2006 9:11 AM 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.83 MB
                (1,920,512 bytes) 3/25/2005 8:00 PM

```

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
winlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	901.00 KB (922,624 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\winlogon.exe
ntdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.20 MB (1,257,472 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ntdll.dll
kernel32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.43 MB (1,500,160 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\kernel32.dll
advapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.00 MB (1,051,136 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\advapi32.dll
rpcrt4	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.63 MB (1,714,176 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\rpcrt4.dll
crypt32	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	1.36 MB (1,428,992 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\crypt32.dll
msasn1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	152.50 KB (156,160 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msasn1.dll
msvcrt	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)	508.00 KB (520,192 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msvcrt.dll
user32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.04 MB (1,085,952 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\user32.dll
gdi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	592.00 KB (606,208 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\gdi32.dll
nddeapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	25.00 KB (25,600 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\nddeapi.dll
profmap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.00 KB (36,864 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\profmap.dll
netapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	589.00 KB (603,136 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netapi32.dll
userenv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.02 MB (1,069,056 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\userenv.dll
psapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.00 KB (29,696 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\psapi.dll
regapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	108.50 KB (111,104 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\regapi.dll
secur32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	120.00 KB (122,880 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\secur32.dll
setupapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.45 MB (1,523,200 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\setupapi.dll
version	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB (28,672 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\version.dll

winsta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	89.00 KB (91,136 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\winsta.dll		
ws2_32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	176.50 KB (180,736 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\ws2_32.dll		
ws2help	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	30.50 KB (31,232 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\ws2help.dll		
msgina	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.14 MB (1,193,472 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\msgina.dll		
shsvcs	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	193.50 KB (198,144 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\shsvcs.dll		
shlwapi	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	606.50 KB (621,056 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\shlwapi.dll		
sfc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	6.00 KB (6,144 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\sfc.dll		
sfc_os	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	183.50 KB (187,904 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\sfc_os.dll		
wintrust	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	297.50 KB (304,640 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\wintrust.dll		
imagehlp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	57.50 KB (58,880 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\imagehlp.dll		
ole32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.43 MB (2,543,616 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\ole32.dll		
comctl32	6.0 (srv03_sp1_rtm.050324-1447)	1.51 MB (1,584,128 bytes)	3/30/2006	6:12
PM	Microsoft Corporation	e:\windows\winsxs\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.3790.1830_x-ww_aced72af\comctl32.dll		
winscard	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	230.00 KB (235,520 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\winscard.dll		
wtsapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.00 KB (29,696 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\wtsapi32.dll		
winmm	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	303.50 KB (310,784 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\winmm.dll		
shell32	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	10.01 MB (10,492,416 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\shell32.dll		
sxs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.91 MB (2,003,968 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\sxs.dll		
wldap32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	390.00 KB (399,360 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\wldap32.dll		
rsaenh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	241.96 KB (247,768 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\rsaenh.dll		
cscdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	151.50 KB (155,136 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\cscdll.dll		
dimsntfy	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB (28,672 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\dimsntfy.dll		
wlnotify	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	148.00 KB (151,552 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\wlnotify.dll		
mpr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	115.00 KB (117,760 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\mpr.dll		
oleaut32	5.2.3790.1830	1.06 MB (1,116,160 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\oleaut32.dll		
winspool	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	247.00 KB (252,928 bytes)	3/25/2005	8:00
PM	Microsoft Corporation	e:\windows\system32\winspool.drv		
comctl32	5.82 (srv03_sp1_rtm.050324-1447)	934.50 KB (956,928 bytes)		

	3/30/2006 6:12 PM	Microsoft Corporation			
	e:\windows\winsxs\amd64_microsoft.windows.common-				
controls_uxtheme	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		494.50 KB	(506,368 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\uxtheme.dll		
clbcatq	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)		865.00 KB	(885,760 bytes)	
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\clbcatq.dll		
comres	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)		779.50 KB	(798,208 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\comres.dll		
wbemprox	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		38.00 KB	(38,912 bytes)	
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wbemprox.dll		
wbemcomn	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		524.00 KB	(536,576 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\wbem\wbemcomn.dll		
xpsp2res	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		2.77 MB	(2,899,456 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\xpsp2res.dll		
wbemsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		58.00 KB	(59,392 bytes)	
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wbemsvc.dll		
fastprox	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		866.50 KB	(887,296 bytes)	
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\fastprox.dll		
msvcp60	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)		919.50 KB	(941,568 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msvcp60.dll		
ntdsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		127.50 KB	(130,560 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ntdsapi.dll		
dnsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		297.50 KB	(304,640 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\dnsapi.dll		
services	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		216.50 KB	(221,696 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\services.exe		
ncobjapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		80.00 KB	(81,920 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ncobjapi.dll		
scesrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		594.50 KB	(608,768 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\scesrv.dll		
authz	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		167.00 KB	(171,008 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\authz.dll		
umpnpgmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		205.00 KB	(209,920 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\umpnpgmgr.dll		
eventlog	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		127.00 KB	(130,048 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\eventlog.dll		
lsass	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		14.00 KB	(14,336 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\lsass.exe			
lsasrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.50 MB	(1,568,256 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\lsasrv.dll		
samlib	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		69.00 KB	(70,656 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\samlib.dll			
samsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.01 MB	(1,059,328 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\samsrv.dll		
cryptdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		47.00 KB	(48,128 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\cryptdll.dll			
msprivs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		47.50 KB	(48,640 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\msprivs.dll			
kerberos	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		698.00 KB	(714,752 bytes)	
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\kerberos.dll		

madv1_0	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	253.00	KB	(259,072	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\madv1_0.dll			
iphlpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	177.00	KB	(181,248	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\iphlpapi.dll			
netlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	666.00	KB	(681,984	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\netlogon.dll			
w32time	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	400.50	KB	(410,112	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\w32time.dll			
schannel	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	248.00	KB	(253,952	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\schannel.dll			
wdigest	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	130.50	KB	(133,632	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\wdigest.dll			
rassfm	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.00	KB	(36,864 bytes)	3/25/2005 8:00
PM	Microsoft Corporation	e:\windows\system32\rassfm.dll			
kdcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	409.00	KB	(418,816	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\kdcsvc.dll			
ntlsa	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.81	MB	(2,948,096	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\ntlsa.dll			
esent	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.26	MB	(2,366,976	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\esent.dll			
ntdsatq	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	51.00	KB	(52,224 bytes)	3/25/2005 8:00
PM	Microsoft Corporation	e:\windows\system32\ntdsatq.dll			
msswsock	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	478.00	KB	(489,472	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\msswsock.dll			
scecli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	308.00	KB	(315,392	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\scecli.dll			
ws03res	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	794.00	KB	(813,056	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\ws03res.dll			
hnetcfg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	561.00	KB	(574,464	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\hnetcfg.dll			
wshtcpip	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.00	KB	(29,696	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\wshtcpip.dll			
ipsecsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	358.50	KB	(367,104	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\ipsecsvc.dll			
oakley	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	372.50	KB	(381,440	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\oakley.dll			
winipsec	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	52.50	KB	(53,760	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\winipsec.dll			
pstorsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.00	KB	(36,864 bytes)	3/25/2005 8:00
PM	Microsoft Corporation	e:\windows\system32\pstorsvc.dll			
psbase	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	124.00	KB	(126,976	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\psbase.dll			
dssenh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	226.96	KB	(232,408	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\dssenh.dll			
wlbcctrl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	137.50	KB	(140,800	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\wlbcctrl.dll			
svchost	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.50	KB	(25,088 bytes)	3/25/2005 8:00
PM	Microsoft Corporation	e:\windows\system32\svchost.exe			
rpcss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	672.00	KB	(688,128	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\rpcss.dll			
ntmarta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	222.50	KB	(227,840	bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\ntmarta.dll			
wzcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	492.00	KB	(503,808	bytes)

	3/25/2005 1:35 AM	Microsoft Corporation	e:\windows\system32\wzcsvc.dll		
rtutils	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		66.00 KB (67,584 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\rtutils.dll		
wmi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		5.50 KB (5,632 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\wmi.dll		
dhcpcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		219.00 KB (224,256 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\dhcpcsvc.dll		
atl	3.05.2284		96.50 KB (98,816 bytes)	3/25/2005 8:00 PM	Microsoft Corporation
			e:\windows\system32\atl.dll		
rastls	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		236.50 KB (242,176 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\rastls.dll		
cryptui	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)		705.50 KB (722,432 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\cryptui.dll		
mprapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		154.50 KB (158,208 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\mprapi.dll		
activeds	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		348.50 KB (356,864 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\activeds.dll		
adslsdp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		240.50 KB (246,272 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\adslsdp.dll		
credui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		202.00 KB (206,848 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\credui.dll		
rasapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		410.00 KB (419,840 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\rasapi32.dll		
rasman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		95.50 KB (97,792 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\rasman.dll		
tapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		332.50 KB (340,480 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\tapi32.dll		
raschap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		141.00 KB (144,384 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\raschap.dll		
schedsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		308.50 KB (315,904 bytes)	3/30/2006 10:25 AM	
		Microsoft Corporation	e:\windows\system32\schedsvc.dll		
msidle	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		9.00 KB (9,216 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\msidle.dll		
wkssvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		221.00 KB (226,304 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\wkssvc.dll		
wiarpc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		57.00 KB (58,368 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\wiarpc.dll		
aelupsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		31.50 KB (32,256 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\aelupsvc.dll		
apphelp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		241.00 KB (246,784 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\apphelp.dll		
cryptsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		114.00 KB (116,736 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\cryptsvc.dll		
certcli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		372.00 KB (380,928 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\certcli.dll		
vssapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.26 MB (1,320,960 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\vssapi.dll		
es	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)		357.00 KB (365,568 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\es.dll		
pchsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		76.00 KB (77,824 bytes)	3/30/2006 10:25 AM	
		Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\pchsvc.dll		
srvsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		156.50 KB (160,256 bytes)	3/25/2005 8:00 PM	
		Microsoft Corporation	e:\windows\system32\srvsvc.dll		

dmserver	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.50	KB	(37,376	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\dmserver.dll		
comsvcs	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)	2.06	MB	(2,156,544	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\comsvcs.dll		
seclogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	27.50	KB	(28,160	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\seclogon.dll		
sens	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	63.50	KB	(65,024 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\sens.dll			
trkwks	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	177.50	KB	(181,760	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\trkwks.dll		
wmisvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	227.00	KB	(232,448	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wmisvc.dll		
wuauerv	5.7.3790.1830 (srv03_sp1_rtm.050324-1447)	12.00	KB	(12,288	bytes)
	3/30/2006 10:25 AM	Microsoft Corporation	e:\windows\system32\wuauerv.dll		
wuaueng	5.7.3790.1830 (srv03_sp1_rtm.050324-1447)	2.17	MB	(2,270,720	bytes)
	3/30/2006 10:25 AM	Microsoft Corporation	e:\windows\system32\wuaueng.dll		
advpack	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	146.00	KB	(149,504	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\advpack.dll		
cabinet	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	138.50	KB	(141,824	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\cabinet.dll		
mspatcha	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	48.00	KB	(49,152	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mspach.dll		
shfolder	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	34.00	KB	(34,816 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\shfolder.dll			
winhttp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	508.50	KB	(520,704	bytes)
	3/30/2006 6:12 PM	Microsoft Corporation	e:\windows\winsxs\amd64_microsoft.windows.winhttp_6595b64144ccf1df_5.1.3790.1830_x-ww_a61ef4db\winhttp.dll		
browser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	125.50	KB	(128,512	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\browser.dll		
netrap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	26.00	KB	(26,624 bytes)	3/25/2005 8:00 PM
	Microsoft Corporation	e:\windows\system32\netrap.dll			
wbemcore	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.24	MB	(1,299,968	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wbemcore.dll		
esscli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	626.50	KB	(641,536	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\esscli.dll		
wmiutils	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	171.00	KB	(175,104	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wmiutils.dll		
repdrvfs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	353.50	KB	(361,984	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\repdrvfs.dll		
wmiprvsd	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	743.00	KB	(760,832	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wmiprvsd.dll		
wbemess	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	532.50	KB	(545,280	bytes)
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\wbem\wbemess.dll		
ncprov	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	73.00	KB	(74,752 bytes)	3/30/2006 10:23 AM
	Microsoft Corporation	e:\windows\system32\wbem\ncprov.dll			
netman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	457.00	KB	(467,968	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netman.dll		
netshell	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.32	MB	(2,437,120	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netshell.dll		
clusapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	127.00	KB	(130,048	bytes)
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\clusapi.dll		
wininet	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.13	MB	(1,186,304	bytes)

	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\wininet.dll				
wzcsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		49.00 KB (50,176 bytes)	3/25/2005	1:35		
AM		Microsoft Corporation	e:\windows\system32\wzcsapi.dll				
rasdlg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		859.50 KB (880,128 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\rasdlg.dll				
rasadhlp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		12.00 KB (12,288 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\rasadhlp.dll				
spoolsv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		107.00 KB (109,568 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\spoolsv.exe				
spoolss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		163.00 KB (166,912 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\spoolss.dll				
localspl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		730.50 KB (748,032 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\localspl.dll				
cnbjmon	5.2.3790.1224 (dnsrv(skatar).040514-1058)		63.00 KB (64,512 bytes)				
	3/25/2005 1:15 AM	Microsoft Corporation	e:\windows\system32\cnbjmon.dll				
pjlmon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		25.50 KB (26,112 bytes)	3/25/2005	1:22		
AM		Microsoft Corporation	e:\windows\system32\pjlmon.dll				
tcpmon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		91.00 KB (93,184 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\tcpmon.dll				
wsnmp32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		67.50 KB (69,120 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\wsnmp32.dll				
tcpmib	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		25.00 KB (25,600 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\tcpmib.dll				
wsock32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		24.50 KB (25,088 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\wsock32.dll				
mgmtapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		22.50 KB (23,040 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mgmtapi.dll				
snmpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		31.50 KB (32,256 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\snmpapi.dll				
usbmon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		28.50 KB (29,184 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\usbmon.dll				
winrnr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		30.00 KB (30,720 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\winrnr.dll				
wshqos	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		33.50 KB (34,304 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\wshqos.dll				
win32spl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		167.00 KB (171,008 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\win32spl.dll				
inetpp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		148.50 KB (152,064 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\inetpp.dll				
icmp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		3.50 KB (3,584 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\icmp.dll				
ersvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		31.00 KB (31,744 bytes)	3/25/2005	8:00		
PM		Microsoft Corporation	e:\windows\system32\ersvc.dll				
msftesql	12.0.5626.1	152.20 KB (155,856 bytes)		8/26/2005	5:17	PM	Microsoft Corporation
			e:\program files\microsoft sql server\mssql.1\mssql\binn\msftesql.exe				
msfte	12.0.5626.1	3.63 MB (3,803,344 bytes)		8/26/2005	5:17	PM	Microsoft Corporation
			e:\program files\microsoft sql server\mssql.1\mssql\binn\msfte.dll				
dbghelp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.22 MB (1,274,368 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\dbghelp.dll				
msftepxy	12.0.5626.1	121.70 KB (124,624 bytes)		8/26/2005	5:17	PM	Microsoft Corporation
			e:\program files\microsoft				
server\mssql.1\mssql\binn\msftepxy.dll							sql
termsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		354.50 KB (363,008 bytes)				

	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\termsrv.dll				
icaapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		27.50 KB (28,160 bytes)	3/30/2006	10:23 AM		
		Microsoft Corporation	e:\windows\system32\icaapi.dll				
mstlsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		187.00 KB (191,488 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mstlsapi.dll				
rdpwsx	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		170.13 KB (174,216 bytes)				
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\rdpwsx.dll				
rdpsnd	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		25.00 KB (25,600 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\rdpsnd.dll				
scredir	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		38.50 KB (39,424 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\scredir.dll				
cscui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		441.00 KB (451,584 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\cscui.dll				
msacm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		31.00 KB (31,744 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msacm32.drv				
msacm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		112.00 KB (114,688 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msacm32.dll				
drprov	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		24.00 KB (24,576 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\drprov.dll				
imaadp32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		24.00 KB (24,576 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\imaadp32.acm				
ntlanman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		71.50 KB (73,216 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ntlanman.dll				
netui0	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		130.00 KB (133,120 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netui0.dll				
netui1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		338.50 KB (346,624 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netui1.dll				
davclnt	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		38.00 KB (38,912 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\davclnt.dll				
msadp32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		23.50 KB (24,064 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msadp32.acm				
msg711	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		13.50 KB (13,824 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\msg711.acm				
mprui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		67.50 KB (69,120 bytes)	3/25/2005	8:00 PM		
		Microsoft Corporation	e:\windows\system32\mprui.dll				
netui2	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		542.00 KB (555,008 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netui2.dll				
comdlg32	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		446.50 KB (457,216 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\comdlg32.dll				
netmsg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		179.00 KB (183,296 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\netmsg.dll				
msgsm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		34.50 KB (35,328 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msgsm32.acm				
tssoft32	1.01		13.50 KB (13,824 bytes)	3/25/2005	8:00 PM	DSP	GROUP, INC.
			e:\windows\system32\tssoft32.acm				
tsd32	1.03		24.50 KB (25,088 bytes)	3/25/2005	8:00 PM	DSP	GROUP, INC.
			e:\windows\system32\tsd32.dll				
printui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		928.50 KB (950,784 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\printui.dll				
cfgmgr32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		18.00 KB (18,432 bytes)				
	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\cfgmgr32.dll				
rdpclip	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		99.00 KB (101,376 bytes)				
	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\rdpclip.exe				

urlmon	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.02 MB	(1,074,176 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\urlmon.dll	
explorer	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.30 MB	(1,364,480 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\explorer.exe	
browseui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.53 MB	(1,601,536 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\browseui.dll	
shdocvw	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	2.30 MB	(2,416,128 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\shdocvw.dll	
themeui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	530.50 KB	(543,232 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\themeui.dll	
msimg32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	6.50 KB	(6,656 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\msimg32.dll	
actxprxy	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	220.50 KB	(225,792 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\actxprxy.dll	
linkinfo	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	30.00 KB	(30,720 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\linkinfo.dll	
ntshrui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	184.00 KB	(188,416 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\ntshrui.dll	
webcheck	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	439.00 KB	(449,536 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\webcheck.dll	
stobject	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	142.50 KB	(145,920 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\stobject.dll	
batmeter	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	41.50 KB	(42,496 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\batmeter.dll	
powrprof	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	32.50 KB	(33,280 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\powrprof.dll	
browsecl	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	63.00 KB	(64,512 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\browsecl.dll	
duser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	445.00 KB	(455,680 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\duser.dll	
mlang	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	686.00 KB	(702,464 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\mlang.dll	
mydocs	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	101.00 KB	(103,424 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\mydocs.dll	
shdoclc	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	589.50 KB	(603,648 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\shdoclc.dll	
zipfldr	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	449.50 KB	(460,288 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\zipfldr.dll	
cmd	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	538.50 KB	(551,424 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\cmd.exe	
helpctr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.30 MB	(1,363,456 bytes)
	3/30/2006 10:25 AM Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\helpctr.exe	
hcappres	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	7.50 KB	(7,680 bytes)
	3/30/2006 10:25 AM Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\hcappres.dll	
itss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	208.00 KB	(212,992 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\itss.dll	
msxml3	8.70.1104.0	2.04 MB	(2,141,184 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\msxml3.dll	
pchshell	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	155.00 KB	(158,720 bytes)
	3/30/2006 10:25 AM Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\pchshell.dll	

mshtml	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	5.65 MB (5,928,448 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mshtml.dll
msls31	3.10.349.0	357.00 KB (365,568 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msls31.dll
msimtf	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	380.50 KB (389,632 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msimtf.dll
msctf	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	617.50 KB (632,320 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\msctf.dll
jscript	5.6.0.8827	974.50 KB (997,888 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\jscript.dll
imm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	208.00 KB (212,992 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\imm32.dll
mshtml	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	905.50 KB (927,232 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mshtml.dll
vbscript	5.6.0.8827	646.50 KB (662,016 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\vbscript.dll
msinfo	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	636.00 KB (651,264 bytes)	3/30/2006 10:25 AM	Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\msinfo.dll
mfc42u	6.50.9146.0	1.39 MB (1,462,272 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mfc42u.dll
riched32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	7.00 KB (7,168 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\riched32.dll
riched20	5.31.23.1224	1.10 MB (1,157,120 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\riched20.dll
helpsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.52 MB (1,591,296 bytes)	3/30/2006 10:25 AM	Microsoft Corporation	e:\windows\pchealth\helpctr\binaries\helpsvc.exe
mmc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.83 MB (1,920,512 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mmc.exe
mmcbase	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	106.50 KB (109,056 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mmcbase.dll
oleacc	4.2.5406.0 (srv03_sp1_rtm.050324-1447)	374.50 KB (383,488 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\oleacc.dll
mmcndmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.23 MB (2,336,256 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mmcndmgr.dll
cmprops	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	313.00 KB (320,512 bytes)	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\cmprops.dll
mmfutil	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	19.50 KB (19,968 bytes)	3/30/2006 10:23 AM	Microsoft Corporation	e:\windows\system32\mmfutil.dll
ntmsmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	912.50 KB (934,400 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ntmsmgr.dll
ntmsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	90.50 KB (92,672 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\ntmsapi.dll
els	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	327.00 KB (334,848 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\els.dll
dfrgsnap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	51.50 KB (52,736 bytes)	3/25/2005 8:00 PM	Microsoft Corp. and Executive Software International, Inc.	e:\windows\system32\dfrgsnap.dll
dfrgres	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	51.50 KB (52,736 bytes)	3/25/2005 8:00 PM	Microsoft Corp. and Executive Software International, Inc.	e:\windows\system32\dfrgres.dll
mycomput	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	176.50 KB (180,736 bytes)	3/25/2005 8:00 PM	Microsoft Corporation	e:\windows\system32\mycomput.dll

filemgmt	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	610.00 KB	(624,640 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\filemgmt.dll	
wbemcntl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	316.50 KB	(324,096 bytes)
	3/30/2006 10:23 AM Microsoft Corporation	e:\windows\system32\wbem\wbemcntl.dll	
localesec	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	536.50 KB	(549,376 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\localesec.dll	
smlogcfg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	711.50 KB	(728,576 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\smlogcfg.dll	
odbc32	3.526.1830.0 (srv03_sp1_rtm.050324-1447)	408.00 KB	(417,792 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\odbc32.dll	
pdh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	563.00 KB	(576,512 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\pdh.dll	
odbcbcpl	2000.086.1830.00 (srv03_sp1_rtm.050324-1447)	32.00 KB	(32,768 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\odbcbcpl.dll	
odbcint	3.526.1830.0 (srv03_sp1_rtm.050324-1447)	96.00 KB	(98,304 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\odbcint.dll	
snmppsnap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	312.50 KB	(320,000 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\snmppsnap.dll	
dmdskmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	349.50 KB	(357,888 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\dmdskmgr.dll	
dmutil	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	57.00 KB	(58,368 bytes)
	3/25/2005 1:16 AM Microsoft Corporation	e:\windows\system32\dmutil.dll	
dmdskres	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	116.50 KB	(119,296 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\dmdskres.dll	
devmgr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	456.00 KB	(466,944 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\devmgr.dll	
rasuser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	278.50 KB	(285,184 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\rasuser.dll	
dsprop	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	213.50 KB	(218,624 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\dsprop.dll	
dsuixt	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	145.50 KB	(148,992 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\dsuixt.dll	
mprsnap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.62 MB	(1,703,424 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\mprsnap.dll	
rtrfiltr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	134.50 KB	(137,728 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\rtrfiltr.dll	
servdeps	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	84.00 KB	(86,016 bytes)
	3/30/2006 10:23 AM Microsoft Corporation	e:\windows\system32\servdeps.dll	
adsnt	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	535.50 KB	(548,352 bytes)
	3/25/2005 8:00 PM Microsoft Corporation	e:\windows\system32\adsnt.dll	

[Services]

Display Name	Name	State	Start Mode	Service Type	Path	Error Control	Start
Name	Tag ID						
Application Experience Lookup Service	AeLookupSvc	Running	Auto	Share	e:\windows\system32\svchost.exe -k netsvcs	Normal LocalSystem	0
Alerter	Alerter	Stopped	Disabled	Share Process	e:\windows\system32\svchost.exe	-k	
Application Layer Gateway Service	ALG	Stopped	Manual	Own			Process
Application Management AppMgmt		Stopped	Manual	Share	e:\windows\system32\svchost.exe -k netsvcs	Normal LocalSystem	0

ASP.NET State Service	aspnet_state	Stopped	Manual	Own				Process
	e:\windows\microsoft.net\framework64\v2.0.50727\aspnet_state.exe							Normal NT
AUTHORITY\NetworkService	0							
Windows Audio	AudioSrv	Stopped	Disabled		Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share				Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
Computer Browser	BrowserRunning		Auto	Share				Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
Indexing Service	CiSvc	Stopped	Disabled		Share	Process	e:\windows\system32\cisvc.exe	
	Normal LocalSystem 0							
ClipBook	ClipSrv	Stopped	Disabled		Own	Process	e:\windows\system32\clipsrv.exe	
	Normal LocalSystem 0							
.NET Runtime Optimization Service	v2.0.50727_X86			clr_optimization_v2.0.50727_32	Stopped			
	Manual Own Process e:\windows\microsoft.net\framework\v2.0.50727\mscorsvw.exe							
	Ignore LocalSystem 0							
.NET Runtime Optimization Service	v2.0.50727_x64			clr_optimization_v2.0.50727_64	Stopped			
	Manual Own Process e:\windows\microsoft.net\framework64\v2.0.50727\mscorsvw.exe							
	Ignore LocalSystem 0							
COM+ System Application	COMSysApp	Stopped	Manual	Own				Process
	e:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}							
	Normal LocalSystem 0							
Cryptographic Services	CryptSvc	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
DCOM Server Process Launcher	DcomLaunch	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k dcomlaunch							Normal LocalSystem 0
Distributed File System	Dfs	Stopped	Manual	Own	Process		e:\windows\system32\dfssvc.exe	
	Normal LocalSystem 0							
DHCP Client	Dhcp	Running		Auto	Share	Process	e:\windows\system32\svchost.exe	
	-k networkservice							Normal NT AUTHORITY\NetworkService 0
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	Share				Process
	e:\windows\system32\dmadmin.exe /com							Normal LocalSystem 0
Logical Disk Manager	dmserver	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
DNS Client	Dnscache	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k networkservice							Normal NT
AUTHORITY\NetworkService	0							
Error Reporting Service	ERSvc	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k winerr							Ignore LocalSystem 0
Event Log	Eventlog	Running		Auto	Share			Process
	e:\windows\system32\services.exe							Normal LocalSystem 0
COM+ Event System	EventSystem	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
Help and Support	helpsvc	Running		Auto	Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
Human Interface Device Access	HidServ	Stopped	Disabled		Share			Process
	e:\windows\system32\svchost.exe -k netsvcs							Normal LocalSystem 0
HTTP SSL	HTTPFilter	Stopped	Manual	Share	Process		e:\windows\system32\lsass.exe	
	Normal LocalSystem 0							
IAS Jet Database Access	IASJet	Stopped	Manual	Share				Process
	e:\windows\syswow64\svchost.exe -k iasjet							Normal LocalSystem 0
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled		Own			Process
	e:\windows\system32\imapi.exe							Normal LocalSystem 0

Intersite Messaging	IsmServ	Stopped	Disabled	Own		Process
e:\windows\system32\ismserv.exe Normal LocalSystem 0						
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share		Process
e:\windows\system32\lsass.exe Normal LocalSystem 0						
Server lanmanserver	Running	Auto	Share	Process	e:\windows\system32\svchost.exe	
-k netsvcs	Normal	LocalSystem	0			
Workstation lanmanworkstation	Running	Auto	Share			Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0						
License Logging LicenseService	Stopped	Disabled	Own			Process
e:\windows\system32\llssrv.exe Normal NT AUTHORITY\NetworkService 0						
TCP/IP NetBIOS Helper LmHosts	Running	Auto	Share			Process
e:\windows\system32\svchost.exe -k localservice Normal NT AUTHORITY\LocalService 0						
Messenger Messenger	Stopped	Disabled	Share			Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0						
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Disabled	Own		Process
e:\windows\system32\mnmsrvc.exe Normal LocalSystem 0						
Distributed Transaction Coordinator	MSDTC	Running	Auto	Own		Process
e:\windows\system32\msdtc.exe Normal NT AUTHORITY\NetworkService 0						
SQL Server Integration Services MsDtsServer	Running	Auto	Own			Process
"e:\program files\microsoft sql server\90\dts\binn\msdtssrvr.exe" Normal NT AUTHORITY\NetworkService 0						
SQL Server FullText Search (MSSQLSERVER)	msftesql	Running	Auto	Own		Process
"e:\program files\microsoft sql server\mssql.1\mssql\binn\msftesql.exe" -s:mssql.1 -f:mssqlserver Normal LocalSystem 0						
Windows Installer MSIServer	Stopped	Manual	Share			Process
e:\windows\system32\msiexec.exe /v Normal LocalSystem 0						
SQL Server (MSSQLSERVER) MSSQLSERVER	Stopped	Manual	Own			Process
"e:\program files\microsoft sql server\mssql.1\mssql\binn\sqlservr.exe" -smssqlserver Normal LocalSystem 0						
SQL Server Active Directory Helper	MSSQLServerADHelper	Stopped	Disabled	Own		Process
"e:\program files\microsoft sql server\90\shared\sqladhlp90.exe" Normal NT AUTHORITY\NetworkService 0						
Network DDE NetDDE	Stopped	Disabled	Share			Process
e:\windows\system32\netdde.exe Normal LocalSystem 0						
Network DDE DSDM NetDDEdsdm	Stopped	Disabled	Share			Process
e:\windows\system32\netdde.exe Normal LocalSystem 0						
Net Logon Netlogon	Stopped	Manual	Share	Process	e:\windows\system32\lsass.exe	
Normal LocalSystem 0						
Network Connections Netman	Running	Manual	Share			Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0						
Network Location Awareness (NLA) Nla	Running	Manual	Share			Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0						
File Replication NtFrs	Stopped	Manual	Own	Process	e:\windows\system32\ntfrs.exe	Ignore
LocalSystem 0						
NT LM Security Support Provider NtLmSsp	Running	Manual	Share			Process
e:\windows\system32\lsass.exe Normal LocalSystem 0						
Removable Storage NtmsSvc	Stopped	Manual	Share			Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0						
Office Source Engine ose	Stopped	Manual	Own	Process	"e:\program files (x86)\common files\microsoft shared\source engine\ose.exe"	
Normal LocalSystem 0						
Plug and Play PlugPlay	Running	Auto	Share			Process
e:\windows\system32\services.exe Normal LocalSystem 0						

IPSEC Services PolicyAgent	Running	Auto	Share		Process
e:\windows\system32\lsass.exe Normal LocalSystem 0					
Protected Storage	ProtectedStorageRunning	Auto	Share		Process
e:\windows\system32\lsass.exe Normal LocalSystem 0					
Remote Access Auto Connection Manager	RasAuto	Stopped	Manual	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Remote Access Connection Manager	RasMan	Stopped	Manual	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Remote Desktop Help Session Manager	RDSessMgr	Stopped	Manual	Own	Process
e:\windows\system32\sessmgr.exe Normal LocalSystem 0					
Routing and Remote Access	RemoteAccess	Stopped	Disabled	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Remote Registry	RemoteRegistry	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k regsvc Normal NT AUTHORITY\LocalService 0					
Remote Procedure Call (RPC) Locator	RpcLocator	Stopped	Manual	Own	Process
e:\windows\system32\locator.exe Normal NT AUTHORITY\NetworkService 0					
Remote Procedure Call (RPC)	RpcSs	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k rpss Normal NT AUTHORITY\NetworkService 0					
Resultant Set of Policy Provider	RSOPProv	Stopped	Manual	Share	Process
e:\windows\system32\rsopprov.exe Normal LocalSystem 0					
Special Administration Console Helper	sacsrv	Stopped	Manual	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Security Accounts Manager	SamSs	Running	Auto	Share	Process
e:\windows\system32\lsass.exe Normal LocalSystem 0					
Smart Card	SCardSvr	Stopped	Manual	Share	Process
e:\windows\system32\scardsvr.exe Ignore NT AUTHORITY\LocalService 0					
Task Scheduler	Schedule	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Secondary Logon	seclogon	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Ignore LocalSystem 0					
System Event Notification	SENS	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Windows Firewall/Internet Connection Sharing (ICS)	Share Process		SharedAccess	Stopped	Disabled
e:\windows\system32\svchost.exe -k netsvcs Normal LocalSystem 0					
Shell Hardware Detection	ShellHWDetection	Running	Auto	Share	Process
e:\windows\system32\svchost.exe -k netsvcs Ignore LocalSystem 0					
Print Spooler	Spooler	Running	Auto	Own Process	e:\windows\system32\spoolsv.exe
Normal LocalSystem 0					
SQL Server Browser	SQLBrowser	Stopped	Disabled	Own Process	"e:\program files (x86)\microsoft sql server\90\shared\sqlbrowser.exe"
Normal LocalSystem 0					
SQL Server Agent (MSSQLSERVER)	SQLSERVERAGENT	Stopped	Manual	Own Process	"e:\program files\microsoft sql server\mssql.1\mssql\binn\sqlagent90.exe" -i mssqlserver
Normal LocalSystem 0					
SQL Server VSS Writer	SQLWriter	Stopped	Manual	Own Process	"e:\program files\microsoft sql server\90\shared\sqlwriter.exe"
Normal LocalSystem 0					
Windows Image Acquisition (WIA)	stisvc	Stopped	Disabled	Share	Process
e:\windows\system32\svchost.exe -k imgsvc Normal NT AUTHORITY\LocalService 0					
Microsoft Software Shadow Copy Provider	swprv	Stopped	Manual	Own Process	Process
e:\windows\system32\svchost.exe -k swprv Normal LocalSystem 0					

Performance Logs and Alerts	SysmonLog	Stopped	Auto	Own		Process
e:\windows\system32\smlogsvc.exe		Normal	NT Authority\NetworkService	0		
Telephony	TapiSrv	Stopped	Manual	Share	Process	
tapisrv	Normal	LocalSystem	0		e:\windows\system32\svchost.exe	-k
Terminal Services	TermService	Running		Manual	Share	Process
e:\windows\system32\svchost.exe		-k termsvcs	Normal	LocalSystem	0	
Themes	Themes	Stopped	Disabled		Share	Process
netstvc	Normal	LocalSystem	0		e:\windows\system32\svchost.exe	-k
Telnet	TlntSvr	Stopped	Disabled		Own	Process
NT AUTHORITY\LocalService		0			e:\windows\system32\tlntsvr.exe	Normal
Distributed Link Tracking Server	TrkSrv	Stopped	Disabled		Share	Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
Distributed Link Tracking Client	TrkWks	Running		Auto	Share	Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
Terminal Services Session Directory	Tssdis	Stopped	Disabled		Own	Process
e:\windows\system32\tssdis.exe		Normal	LocalSystem	0		
Windows User Mode Driver Framework	UMWdf	Stopped	Manual	Own		Process
e:\windows\system32\wdfmgr.exe		Normal	NT AUTHORITY\LocalService	0		
Uninterruptible Power Supply	UPS	Stopped	Manual	Own		Process
e:\windows\system32\ups.exe		Normal	NT AUTHORITY\LocalService	0		
Virtual Disk Service	vds	Stopped	Manual	Own	Process	
Normal		LocalSystem	0		e:\windows\system32\vds.exe	
Volume Shadow Copy	VSS	Stopped	Manual	Own	Process	
Normal		LocalSystem	0		e:\windows\system32\vssvc.exe	
Windows Time	W32Time	Running		Auto	Share	Process
e:\windows\system32\svchost.exe		-k localservice	Normal	NT AUTHORITY\LocalService	0	
WebClient	WebClient	Stopped	Disabled		Share	Process
e:\windows\system32\svchost.exe		-k localservice	Normal	NT AUTHORITY\LocalService	0	
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share		Process
e:\windows\system32\svchost.exe		-k localservice	Normal	NT AUTHORITY\LocalService	0	
Windows Management Instrumentation	winmgmt	Running		Auto	Share	Process
e:\windows\system32\svchost.exe		-k netsvcs	Ignore	LocalSystem	0	
Portable Media Serial Number Service	WmdmPmSN	Stopped	Manual	Share		Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
Windows Management Instrumentation Driver Extensions	Wmi	Stopped	Manual	Share		Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own		Process
e:\windows\system32\wbem\wmiapsrv.exe		Normal	LocalSystem	0		
Automatic Updates	wuauclt	Running		Auto	Share	Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
Wireless Configuration	WZC	Running		Auto	Share	Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	
Network Provisioning Service	xmlprov	Stopped	Manual	Share		Process
e:\windows\system32\svchost.exe		-k netsvcs	Normal	LocalSystem	0	

[Program Groups]

Group Name	Name	User Name
Accessories	Default User:Accessories	Default User
Accessories\Accessibility	Default User:Accessories\Accessibility	Default User

Accessories\Entertainment	Default User:Accessories\Entertainment	Default User
Startup	Default User:Startup	Default User
Accessories	All Users:Accessories	All Users
Accessories\Accessibility	All Users:Accessories\Accessibility	All Users
Accessories\Communications	All Users:Accessories\Communications	All Users
Accessories\Entertainment	All Users:Accessories\Entertainment	All Users
Accessories\System Tools	All Users:Accessories\System Tools	All Users
Administrative Tools	All Users:Administrative Tools	All Users
Microsoft SQL Server 2005	All Users:Microsoft SQL Server 2005	All Users
Microsoft SQL Server 2005\Analysis Services	All Users:Microsoft SQL Server 2005\Analysis Services	All Users
Microsoft SQL Server 2005\Configuration Tools	All Users:Microsoft SQL Server 2005\Configuration Tools	All Users
Microsoft SQL Server 2005\Documentation and Tutorials	All Users:Microsoft SQL Server 2005\Documentation and Tutorials	All Users
Microsoft SQL Server 2005\Documentation and Tutorials\Tutorials	All Users:Microsoft SQL Server 2005\Documentation and Tutorials\Tutorials	All Users
Microsoft SQL Server 2005\Performance Tools	All Users:Microsoft SQL Server 2005\Performance Tools	All Users
Microsoft Visual Studio 2005	All Users:Microsoft Visual Studio 2005	All Users
Microsoft Visual Studio 2005\Visual Studio Tools	All Users:Microsoft Visual Studio 2005\Visual Studio Tools	All Users
Startup	All Users:Startup	All Users
Accessories	NT AUTHORITY\SYSTEM:Accessories	NT AUTHORITY\SYSTEM
Accessories\Accessibility	NT AUTHORITY\SYSTEM:Accessories\Accessibility	NT AUTHORITY\SYSTEM
Accessories\Entertainment	NT AUTHORITY\SYSTEM:Accessories\Entertainment	NT AUTHORITY\SYSTEM
Startup	NT AUTHORITY\SYSTEM:Startup	NT AUTHORITY\SYSTEM
Accessories	TPCCX64\Administrator:Accessories	TPCCX64\Administrator
Accessories\Accessibility	TPCCX64\Administrator:Accessories\Accessibility	TPCCX64\Administrator
Accessories\Entertainment	TPCCX64\Administrator:Accessories\Entertainment	TPCCX64\Administrator
Administrative Tools	TPCCX64\Administrator:Administrative Tools	TPCCX64\Administrator
Startup	TPCCX64\Administrator:Startup	TPCCX64\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	TPCCX64\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common Startup

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid
Sound	Not Available

Media Clip Not Available
WordPad Document "%programfiles%\windows nt\accessories\wordpad.exe"
Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
[Summary]

Item Value
Version 6.0.3790.1830
Build 63790.1830
Application Path E:\Program Files\Internet Explorer
Language English (United States)
Active Printer Not Available

Cipher Strength 128-bit
Content Advisor Disabled
IEAK Install No

[File Versions]

File	Version	Size	Date	Path	Company		
actxprxy.dll	6.0.3790.1830			221 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
advpack.dll	6.0.3790.1830			146 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
asctrls.ocx	6.0.3790.1830			147 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
browsecl.dll	6.0.3790.1830			63 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
browseui.dll	6.0.3790.1830			1,564 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
cdfview.dll	6.0.3790.1830			216 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
comctl32.dll	5.82.3790.1830			935 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
dxttrans.dll	6.3.3790.1830			320 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
dxtmsft.dll	6.3.3790.1830			549 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						
iecont.dll	<File Missing>			Not Available	Not Available	Not Available	Not Available
iecontlc.dll	<File Missing>			Not Available	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.1830			417 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	
	Microsoft Corporation						

iepeers.dll	6.0.3790.1830	361 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
iesetup.dll	6.0.3790.1830	71 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
ieunit.inf	Not Available	24 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Not Available
ieexplore.exe	6.0.3790.1830	94 KB	3/25/2005 8:00:00 PM	E:\Program Files\Internet Explorer	Microsoft Corporation
imgutil.dll	6.0.3790.1830	61 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
inetcp.cpl	6.0.3790.1830	428 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
inetcp.cpl	6.0.3790.1830	110 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
inseng.dll	6.0.3790.1830	147 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mlang.dll	6.0.3790.1830	686 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
msencode.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
mshhta.exe	6.0.3790.1830	38 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mshhtml.dll	6.0.3790.1830	5,790 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mshhtml.tlb	6.0.3790.1830	1,320 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mshhtml.dll	6.0.3790.1830	906 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mshhtml.dll	6.0.3790.1830	56 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
msident.dll	6.0.3790.1830	69 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
msident.dll	6.0.3790.1830	16 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
msieftp.dll	6.0.3790.1830	369 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
msrating.dll	6.0.3790.1830	240 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
mstime.dll	6.0.3790.1830	878 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
occache.dll	6.0.3790.1830	126 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
proctexe.ocx	<File Missing>	Not Available	Not Available	Not Available	Not Available
sendmail.dll	6.0.3790.1830	64 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
shdocl.dll	6.0.3790.1830	590 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
shdocvw.dll	6.0.3790.1830	2,360 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
shfolder.dll	6.0.3790.1830	34 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
shlwapi.dll	6.0.3790.1830	607 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation
tdc.ocx	1.3.0.3130	91 KB	3/25/2005 8:00:00 PM	E:\WINDOWS\system32	Microsoft Corporation

Corporation
url.dll 6.0.3790.1830 40 KB 3/25/2005 8:00:00 PM E:\WINDOWS\system32 Microsoft Corporation
urlmon.dll 6.0.3790.1830 1,049 KB 3/25/2005 8:00:00 PM E:\WINDOWS\system32 Microsoft Corporation
webcheck.dll 6.0.3790.1830 439 KB 3/25/2005 8:00:00 PM E:\WINDOWS\system32 Microsoft Corporation
wininet.dll 6.0.3790.1830 1,159 KB 3/25/2005 8:00:00 PM E:\WINDOWS\system32 Microsoft Corporation

[Connectivity]

Item Value
Connection Preference Never dial

LAN Settings

AutoConfigProxy wininet.dll
AutoProxyDetectMode Enabled
AutoConfigURL
Proxy Disabled
ProxyServer
ProxyOverride

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item Value
Page Refresh Type Automatic
Temporary Internet Files Folder E:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space Not Available
Available Disk Space Not Available
Maximum Cache Size Not Available
Available Cache Size Not Available

[List of Objects]

Program File Status CodeBase
No cached object information available

[Content]

[Following are sub-categories of this main category]
[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranet Custom
Trusted sites Custom
Internet High
Restricted sites Custom

Client Configuration Parameters

Following is the System Information report for Client 1. Clients 2, 3 and 4 were configured similarly to Client 1.

Microsoft Windows Server 2003 Standard Edition SP1 Client System Information Report

System Information report written at: 04/04/06 21:06:37

System Name: WEB171

[System Summary]

Item	Value
OS Name	Microsoft(R) Windows(R) Server 2003, Standard Edition
Version	5.2.3790 Service Pack 1 Build 3790
Other OS Description	Not Available
OS Manufacturer	Microsoft Corporation
System Name	WEB171
System Manufacturer	Intel
System Model	SJR2A072
System Type	X86-based PC
Processor	x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3591 Mhz
Processor	x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3591 Mhz
Processor	x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3591 Mhz
Processor	x86 Family 15 Model 4 Stepping 1 GenuineIntel ~3591 Mhz
BIOS Version/Date	Intel Coporation SE7520JR23.86B.P.07.00.0072.012720051918, 1/27/2005
SMBIOS Version	2.3
Windows Directory	D:\WINDOWS
System Directory	D:\WINDOWS\system32
Boot Device	\Device\HarddiskVolume1
Locale	United States
Hardware Abstraction Layer	Version = "5.2.3790.1830 (srv03_sp1_rtm.050324-1447)"

User Name WEB171\Administrator
Time Zone China Standard Time
Total Physical Memory 2,047.39 MB
Available Physical Memory 1.64 GB
Total Virtual Memory 3.90 GB
Available Virtual Memory 3.65 GB
Page File Space 2.00 GB
Page File D:\pagefile.sys

[Hardware Resources]

[Conflicts/Sharing]

Resource	Device
I/O Port 0x00000000-0x00000CF7	PCI bus
I/O Port 0x00000000-0x00000CF7	Direct memory access controller

IRQ 16 Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595
IRQ 16 Standard Universal PCI to USB Host Controller

IRQ 18 Standard Universal PCI to USB Host Controller
IRQ 18 Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1

Memory Address 0xA0000-0xBFFFF	PCI bus
Memory Address 0xA0000-0xBFFFF	RAGE XL PCI Family (Microsoft Corporation)

I/O Port 0x0000D000-0x0000DFFF	Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595
I/O Port 0x0000D000-0x0000DFFF	Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A

[DMA]

Resource	Device	Status
Channel 4	Direct memory access controller	OK

[Forced Hardware]

Device PNP Device ID

[I/O]

Resource	Device	Status
0x00000000-0x00000CF7	PCI bus	OK
0x00000000-0x00000CF7	Direct memory access controller	OK
0x00000D00-0x0000FFFF	PCI bus	OK
0x0000D000-0x0000DFFF	Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595	OK
0x0000D000-0x0000DFFF	Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A	OK
0x0000D880-0x0000D8BF	Intel(R) PRO/1000 MT Dual Port Network Connection	OK
0x0000DC00-0x0000DC3F	Intel(R) PRO/1000 MT Dual Port Network Connection #2	OK

0x0000C800-0x0000C81F	Standard Universal PCI to USB Host Controller	OK
0x0000C880-0x0000C89F	Standard Universal PCI to USB Host Controller	OK
0x0000CC00-0x0000CC1F	Standard Universal PCI to USB Host Controller	OK
0x0000E800-0x0000E8FF	RAGE XL PCI Family (Microsoft Corporation)	OK
0x000003B0-0x000003BB	RAGE XL PCI Family (Microsoft Corporation)	OK
0x000003C0-0x000003DF	RAGE XL PCI Family (Microsoft Corporation)	OK
0x00000A79-0x00000A79	ISAPNP Read Data Port	OK
0x00000279-0x00000279	ISAPNP Read Data Port	OK
0x00000274-0x00000277	ISAPNP Read Data Port	OK
0x00000020-0x00000021	Programmable interrupt controller	OK
0x000000A0-0x000000A1	Programmable interrupt controller	OK
0x00000081-0x00000083	Direct memory access controller	OK
0x00000087-0x00000087	Direct memory access controller	OK
0x00000089-0x0000008B	Direct memory access controller	OK
0x0000008F-0x0000008F	Direct memory access controller	OK
0x000000C0-0x000000DF	Direct memory access controller	OK
0x00000040-0x00000043	System timer	OK
0x00000070-0x00000071	System CMOS/real time clock	OK
0x00000060-0x00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x00000064-0x00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
0x00000061-0x00000061	System speaker	OK
0x000000F0-0x000000FF	Numeric data processor	OK
0x000003F8-0x000003FF	Communications Port (COM1)	OK
0x000002F8-0x000002FF	Communications Port (COM2)	OK
0x00000680-0x0000069F	Motherboard resources	OK
0x00000640-0x0000065F	Motherboard resources	OK
0x00000600-0x0000060F	Motherboard resources	OK
0x000006C0-0x000006DF	Motherboard resources	OK
0x00000700-0x0000071F	Motherboard resources	OK
0x00000720-0x0000073F	Motherboard resources	OK
0x00000740-0x0000074F	Motherboard resources	OK
0x000006AE-0x000006AE	Motherboard resources	OK
0x000006AF-0x000006AF	Motherboard resources	OK
0x00000010-0x0000001F	Motherboard resources	OK
0x00000022-0x0000003F	Motherboard resources	OK
0x00000044-0x0000005F	Motherboard resources	OK
0x00000062-0x00000063	Motherboard resources	OK
0x00000065-0x0000006F	Motherboard resources	OK
0x00000072-0x0000007F	Motherboard resources	OK
0x00000080-0x00000080	Motherboard resources	OK
0x00000084-0x00000086	Motherboard resources	OK
0x00000088-0x00000088	Motherboard resources	OK
0x0000008C-0x0000008E	Motherboard resources	OK
0x00000090-0x0000009F	Motherboard resources	OK
0x000000A2-0x000000BF	Motherboard resources	OK
0x000000E0-0x000000EF	Motherboard resources	OK
0x000004D0-0x000004D1	Motherboard resources	OK
0x00000400-0x0000047F	Motherboard resources	OK
0x00000500-0x0000053F	Motherboard resources	OK
0x0000FC00-0x0000FC0F	Intel(R) 82801EB Ultra ATA Storage Controllers - 24DB	OK
0x000001F0-0x000001F7	Primary IDE Channel	OK
0x000003F6-0x000003F6	Primary IDE Channel	OK
0x00000170-0x00000177	Secondary IDE Channel	OK

0x00000376-0x00000376	Secondary IDE Channel	OK
0x0000C480-0x0000C487	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
0x0000C400-0x0000C403	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
0x0000C080-0x0000C087	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
0x0000C000-0x0000C003	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
0x0000BC00-0x0000BC0F	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
0x00000540-0x0000055F	Intel(R) 82801EB SMBus Controller - 24D3	OK

[IRQs]

Resource	Device	Status
IRQ 9	Microsoft ACPI-Compliant System	OK
IRQ 16	Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595	OK
IRQ 16	Standard Universal PCI to USB Host Controller	OK
IRQ 54	Intel(R) PRO/1000 MT Dual Port Network Connection	OK
IRQ 55	Intel(R) PRO/1000 MT Dual Port Network Connection #2	OK
IRQ 19	Standard Universal PCI to USB Host Controller	OK
IRQ 18	Standard Universal PCI to USB Host Controller	OK
IRQ 18	Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	OK
IRQ 23	Standard Enhanced PCI to USB Host Controller	OK
IRQ 17	RAGE XL PCI Family (Microsoft Corporation)	OK
IRQ 0	System timer	OK
IRQ 8	System CMOS/real time clock	OK
IRQ 1	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	OK
IRQ 12	Microsoft PS/2 Mouse	OK
IRQ 13	Numeric data processor	OK
IRQ 4	Communications Port (COM1)	OK
IRQ 3	Communications Port (COM2)	OK
IRQ 14	Primary IDE Channel	OK
IRQ 15	Secondary IDE Channel	OK
IRQ 11	Intel(R) 82801EB SMBus Controller - 24D3	OK

[Memory]

Resource	Device	Status
0xA0000-0xBFFFF	PCI bus	OK
0xA0000-0xBFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0x80000000-0xFFBFFFFF	PCI bus	OK
0xFCE00000-0xFCFFFFFF	Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595	OK
0xFCEFE000-0xFCEFEFFF	Intel(R) 6700PXH I/OxAPIC Interrupt Controller A - 0326	OK
0xFCF00000-0xFCFFFFFF	Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A	OK
0xFCFA0000-0xFCFBFFFF	Intel(R) PRO/1000 MT Dual Port Network Connection	OK
0xFCFE0000-0xFCFFFFFF	Intel(R) PRO/1000 MT Dual Port Network Connection #2	OK
0xFCEFF000-0xFCEFFFFF	Intel(R) 6700PXH I/OxAPIC Interrupt Controller B - 0327	OK
0xFCDFEC00-0xFCDFEFFF	Standard Enhanced PCI to USB Host Controller	OK
0xFD000000-0xFDFFFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xFEBFF000-0xFEBFFFFF	RAGE XL PCI Family (Microsoft Corporation)	OK
0xFED20000-0xFED8FFFF	Motherboard resources	OK
0xFEC00000-0xFEC0FFFF	Motherboard resources	OK
0xFEE00000-0xFEE0FFFF	Motherboard resources	OK
0xFFBFFC00-0xFFBFFFFF	Intel(R) 82801EB Ultra ATA Storage Controllers - 24DB	OK
0xE0000000-0xEFFFFFFF	Motherboard resources	OK

0x0000-0x9FFFF System board OK
 0xC0000-0xDFFFF System board OK
 0xE0000-0xFFFFF System board OK
 0x100000-0x7FFFFFFF System board OK
 0xFFC00000-0xFFFFFFFF System board OK

[Components]

[Multimedia]

[Audio Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
d:\windows\system32\sl_anet.acm	Sipro Lab Telecom Inc.	Sipro Lab Telecom Audio Codec	OK	D:\WINDOWS\system32\SL_ANET.ACM	3.02	84.00 KB (86,016 bytes)	3/25/2003 8:00 PM
d:\windows\system32\msaud32.acm	Microsoft Corporation	Windows Media Audio Codec	OK	D:\WINDOWS\system32\MSAUD32.ACM	8.00.00.4487	288.00 KB (294,912 bytes)	3/25/2003 8:00 PM
d:\windows\system32\tsoft32.acm	DSP GROUP, INC.		OK	D:\WINDOWS\system32\TSSOFT32.ACM	1.01	9.50 KB (9,728 bytes)	3/25/2003 8:00 PM
d:\windows\system32\imaadp32.acm	Microsoft Corporation		OK	D:\WINDOWS\system32\IMAADP32.ACM	5.2.3790.0	15.50 KB (15,872 bytes)	3/25/2003 8:00 PM
d:\windows\system32\msadp32.acm	Microsoft Corporation		OK	D:\WINDOWS\system32\MSADP32.ACM	5.2.3790.0	14.50 KB (14,848 bytes)	3/25/2003 8:00 PM
d:\windows\system32\msg711.acm	Microsoft Corporation		OK	D:\WINDOWS\system32\MSG711.ACM	5.2.3790.0	10.00 KB (10,240 bytes)	3/25/2003 8:00 PM
d:\windows\system32\l3codeca.acm	Fraunhofer Institut Integrierte Schaltungen IIS	Fraunhofer IIS MPEG Layer-3 Codec	OK	D:\WINDOWS\system32\L3CODECA.ACM	1, 9, 0, 0305	284.00 KB (290,816 bytes)	3/25/2003 8:00 PM
d:\windows\system32\msg723.acm	Microsoft Corporation		OK	D:\WINDOWS\system32\MSG723.ACM	5.2.3790.1830	120.00 KB (122,880 bytes)	11/22/2005 4:21 PM
d:\windows\system32\msgsm32.acm	Microsoft Corporation		OK	D:\WINDOWS\system32\MSGSM32.ACM	5.2.3790.0	20.50 KB (20,992 bytes)	3/25/2003 8:00 PM

[Video Codecs]

CODEC	Manufacturer	Description	Status	File	Version	Size	Creation Date
d:\windows\system32\msh261.drv	Microsoft Corporation		OK	D:\WINDOWS\system32\MSH261.DRV	5.2.3790.1830	184.00 KB (188,416 bytes)	11/22/2005 4:21 PM
d:\windows\system32\msrle32.dll	Microsoft Corporation		OK				

D:\WINDOWS\system32\MSRLE32.DLL 5.2.3790.0 (srv03_rtm.030324-2048) 10.50
 KB (10,752 bytes) 3/25/2003 8:00 PM
 d:\windows\system32\msyuv.dll Microsoft Corporation OK
 D:\WINDOWS\system32\MSYUV.DLL 5.2.3790.0 (srv03_rtm.030324-2048) 16.50
 KB (16,896 bytes) 3/25/2003 9:49 AM
 d:\windows\system32\msvide32.dll Microsoft Corporation OK
 D:\WINDOWS\system32\MSVIDC32.DLL 5.2.3790.0 (srv03_rtm.030324-2048)
 26.50 KB (27,136 bytes) 3/25/2003 8:00 PM
 d:\windows\system32\msh263.drv Microsoft Corporation OK
 D:\WINDOWS\system32\MSH263.DRV 5.2.3790.1830 288.00 KB (294,912 bytes)
 11/22/2005 4:21 PM
 d:\windows\system32\iyuv_32.dll Microsoft Corporation OK
 D:\WINDOWS\system32\IYUV_32.DLL 5.2.3790.1830 (srv03_sp1_rtm.050324-1447)
 46.50 KB (47,616 bytes) 11/22/2005 4:21 PM
 d:\windows\system32\tsbyuv.dll Microsoft Corporation OK
 D:\WINDOWS\system32\TSBYUV.DLL 5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB
 (8,192 bytes) 3/25/2003 9:50 AM

[CD-ROM]

Item Value
 Drive F:
 Description CD-ROM Drive
 Media Loaded No
 Media Type CD-ROM
 Name MITSUMI CD-ROM SR244W
 Manufacturer (Standard CD-ROM drives)
 Status OK
 Transfer Rate Not Available
 SCSI Target ID 1
 PNP Device ID IDE\CDROMMITSUMI_CD-
 ROM_SR244W_____T01A_____ \5&20AE57A1&0&0.1.0
 Driver d:\windows\system32\drivers\cdrom.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
 51.00 KB (52,224 bytes), 3/25/2003 8:00 PM)

[Sound Device]

Item Value

[Display]

Item Value
 Name RAGE XL PCI Family (Microsoft Corporation)
 PNP Device ID PCI\VEN_1002&DEV_4752&SUBSYS_10798086&REV_27\4&2E98101C&0&60F0 ID
 Adapter Type ATI RAGE XL PCI (B41), ATI Technologies Inc. compatible
 Adapter Description RAGE XL PCI Family (Microsoft Corporation)
 Adapter RAM 8.00 MB (8,388,608 bytes)
 Installed Drivers ati2drad.dll
 Driver Version 5.10.3663.6013
 INF File atiixpad.inf (ati2mpad section)
 Color Planes 1
 Color Table Entries 4294967296

Resolution 1024 x 768 x 60 hertz
Bits/Pixel 32
Memory Address 0xFD000000-0xFDFFFFFF
I/O Port 0x0000E800-0x0000E8FF
Memory Address 0xFEBFF000-0xFEBFFFFFF
IRQ Channel IRQ 17
I/O Port 0x000003B0-0x000003BB
I/O Port 0x000003C0-0x000003DF
Memory Address 0xA0000-0xBFFFF
Driver d:\windows\system32\drivers\ati2mpad.sys (5.10.3663.6013, 335.38 KB (343,424 bytes),
6/3/2005 10:24 PM)

[Infrared]

Item Value

[Input]

[Keyboard]

Item Value
Description Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
Name Enhanced (101- or 102-key)
Layout 00000409
PNP Device ID ACPI\PNP0303\4&35F762C4&0
Number of Function Keys 12
I/O Port 0x00000060-0x00000060
I/O Port 0x00000064-0x00000064
IRQ Channel IRQ 1
Driver d:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
54.50 KB (55,808 bytes), 3/25/2003 8:00 PM)

[Pointing Device]

Item Value
Hardware Type Microsoft PS/2 Mouse
Number of Buttons 5
Status OK
PNP Device ID ACPI\PNP0F03\4&35F762C4&0
Power Management Supported No
Double Click Threshold 6
Handedness Right Handed Operation
IRQ Channel IRQ 12
Driver d:\windows\system32\drivers\i8042prt.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
54.50 KB (55,808 bytes), 3/25/2003 8:00 PM)

[Modem]

Item Value

[Network]

[Adapter]

Item Value
Name [00000001] RAS Async Adapter
Adapter Type Not Available
Product Type RAS Async Adapter
Installed Yes
PNP Device ID Not Available
Last Reset 4/4/2006 11:18 AM
Index 1
Service Name AsyncMac
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available

Name [00000002] WAN Miniport (L2TP)
Adapter Type Not Available
Product Type WAN Miniport (L2TP)
Installed Yes
PNP Device ID ROOT\MS_L2TPMINIPOINT\0000
Last Reset 4/4/2006 11:18 AM
Index 2
Service Name Rasl2tp
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver d:\windows\system32\drivers\rasl2tp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
66.00 KB (67,584 bytes), 3/25/2003 8:00 PM)

Name [00000003] WAN Miniport (PPTP)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPTP)
Installed Yes
PNP Device ID ROOT\MS_PPTPMINIPOINT\0000
Last Reset 4/4/2006 11:18 AM
Index 3
Service Name PptpMiniport
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available

DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 50:50:54:50:30:30
Driver d:\windows\system32\drivers\rasppptp.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
61.00 KB (62,464 bytes), 3/25/2003 8:00 PM)

Name [00000004] WAN Miniport (PPPOE)
Adapter Type Wide Area Network (WAN)
Product Type WAN Miniport (PPPOE)
Installed Yes
PNP Device ID ROOT\MS_PPPOEMINIPOINT\0000
Last Reset 4/4/2006 11:18 AM
Index 4
Service Name RasPppoe
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 33:50:6F:45:30:30
Driver d:\windows\system32\drivers\raspppoe.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447),
40.00 KB (40,960 bytes), 3/25/2003 8:00 PM)

Name [00000005] Direct Parallel
Adapter Type Not Available
Product Type Direct Parallel
Installed Yes
PNP Device ID ROOT\MS_PTMINIPOINT\0000
Last Reset 4/4/2006 11:18 AM
Index 5
Service Name Raspti
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver d:\windows\system32\drivers\raspti.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 19.50
KB (19,968 bytes), 3/25/2003 8:00 PM)

Name [00000006] WAN Miniport (IP)
Adapter Type Not Available
Product Type WAN Miniport (IP)
Installed Yes
PNP Device ID ROOT\MS_NDISWANIP\0000
Last Reset 4/4/2006 11:18 AM
Index 6

Service Name NdisWan
IP Address Not Available
IP Subnet Not Available
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address Not Available
Driver d:\windows\system32\drivers\ndiswan.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 91.00 KB (93,184 bytes), 3/25/2003 8:00 PM)

Name [00000007] Intel(R) PRO/1000 MT Dual Port Network Connection

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_10798086&REV_03\5&CBC31F2&0&200210
Last Reset 4/4/2006 11:18 AM
Index 7
Service Name E1000
IP Address 192.168.0.171
IP Subnet 255.255.255.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:23:B0:1D:36
Memory Address 0xFCFA0000-0xFCFBFFFF
I/O Port 0x0000D880-0x0000D8BF
IRQ Channel IRQ 54
Driver d:\windows\system32\drivers\e1000325.sys (7.4.19.0 built by: WinDDK, 128.50 KB (131,584 bytes), 3/12/2004 11:45 AM)

Name [00000008] Intel(R) PRO/1000 MT Dual Port Network Connection

Adapter Type Ethernet 802.3
Product Type Intel(R) PRO/1000 MT Dual Port Network Connection
Installed Yes
PNP Device ID
PCI\VEN_8086&DEV_1079&SUBSYS_10798086&REV_03\5&CBC31F2&0&210210
Last Reset 4/4/2006 11:18 AM
Index 8
Service Name E1000
IP Address 0.0.0.0
IP Subnet 0.0.0.0
Default IP Gateway Not Available
DHCP Enabled No
DHCP Server Not Available
DHCP Lease Expires Not Available
DHCP Lease Obtained Not Available
MAC Address 00:04:23:B0:1D:37
Memory Address 0xFCFE0000-0xFCFFFFFF

I/O Port 0x0000DC00-0x0000DC3F
IRQ Channel IRQ 55
Driver d:\windows\system32\drivers\le1000325.sys (7.4.19.0 built by: WinDDK, 128.50 KB
(131,584 bytes), 3/12/2004 11:45 AM)

[Protocol]

Item	Value
Name	MSAFD Tcpip [TCP/IP]
Connectionless Service	No
Guarantees Delivery	Yes
Guarantees Sequencing	Yes
Maximum Address Size	16 bytes
Maximum Message Size	0 bytes
Message Oriented	No
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	No
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	Yes
Supports Graceful Closing	Yes
Supports Guaranteed Bandwidth	No
Supports Multicasting	No

Name	MSAFD Tcpip [UDP/IP]
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No
Supports Broadcasting	Yes
Supports Connect Data	No
Supports Disconnect Data	No
Supports Encryption	No
Supports Expedited Data	No
Supports Graceful Closing	No
Supports Guaranteed Bandwidth	No
Supports Multicasting	Yes

Name	RSVP UDP Service Provider
Connectionless Service	Yes
Guarantees Delivery	No
Guarantees Sequencing	No
Maximum Address Size	16 bytes
Maximum Message Size	63.93 KB (65,467 bytes)
Message Oriented	Yes
Minimum Address Size	16 bytes
Pseudo Stream Oriented	No

Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting Yes

Name RSVP TCP Service Provider
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 16 bytes
Maximum Message Size 0 bytes
Message Oriented No
Minimum Address Size 16 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption Yes
Supports Expedited Data Yes
Supports Graceful Closing Yes
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{3B189DD0-4B65-4D1E-9B42-4CBE50982252}]
SEQPACKET 3
Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{3B189DD0-4B65-4D1E-9B42-4CBE50982252}]
DATAGRAM 3
Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)

Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{E77AE24C-11BF-4916-858C-B3B3F0A6042D}]
SEQPACKET 0

Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{E77AE24C-11BF-4916-858C-B3B3F0A6042D}]
DATAGRAM 0

Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{6DE3072E-A1B9-4056-A8D7-1C34B32195B8}]
SEQPACKET 1

Connectionless Service No

Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{6DE3072E-A1B9-4056-A8D7-1C34B32195B8}]
DATAGRAM 1

Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D55007DD-4956-4EDF-8613-FA05128F3C3E}]
SEQPACKET 2

Connectionless Service No
Guarantees Delivery Yes
Guarantees Sequencing Yes
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting No
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

Name MSAFD NetBIOS [\Device\NetBT_Tcpip_{D55007DD-4956-4EDF-8613-FA05128F3C3E}]
DATAGRAM 2

Connectionless Service Yes
Guarantees Delivery No
Guarantees Sequencing No
Maximum Address Size 20 bytes
Maximum Message Size 62.50 KB (64,000 bytes)
Message Oriented Yes
Minimum Address Size 20 bytes
Pseudo Stream Oriented No
Supports Broadcasting Yes
Supports Connect Data No
Supports Disconnect Data No
Supports Encryption No
Supports Expedited Data No
Supports Graceful Closing No
Supports Guaranteed Bandwidth No
Supports Multicasting No

[WinSock]

Item	Value
File	d:\windows\system32\winsock.dll
Size	2.80 KB (2,864 bytes)
Version	3.10
File	d:\windows\system32\wsock32.dll
Size	22.00 KB (22,528 bytes)
Version	5.2.3790.0 (srv03_rtm.030324-2048)

[Ports]

[Serial]

Item	Value
Name	Communications Port (COM1)
Status	OK
PNP Device ID	ACPI\PNP0501\1
Maximum Input Buffer Size	0
Maximum Output Buffer Size	No
Settable Baud Rate	Yes
Settable Data Bits	Yes
Settable Flow Control	Yes
Settable Parity	Yes
Settable Parity Check	Yes
Settable Stop Bits	Yes
Settable RLSD	Yes
Supports RLSD	Yes
Supports 16 Bit Mode	No
Supports Special Characters	No

Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No
 CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000003F8-0x000003FF
 IRQ Channel IRQ 4
 Driver d:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 64.00
 KB (65,536 bytes), 3/25/2003 8:00 PM)

Name Communications Port (COM2)
 Status OK
 PNP Device ID ACPI\PNP0501\2
 Maximum Input Buffer Size 0
 Maximum Output Buffer Size No
 Settable Baud Rate Yes
 Settable Data Bits Yes
 Settable Flow Control Yes
 Settable Parity Yes
 Settable Parity Check Yes
 Settable Stop Bits Yes
 Settable RLSD Yes
 Supports RLSD Yes
 Supports 16 Bit Mode No
 Supports Special Characters No
 Baud Rate 9600
 Bits/Byte 8
 Stop Bits 1
 Parity None
 Busy No
 Abort Read/Write on Error No
 Binary Mode Enabled Yes
 Continue XMit on XOff No

CTS Outflow Control No
 Discard NULL Bytes No
 DSR Outflow Control 0
 DSR Sensitivity 0
 DTR Flow Control Type Enable
 EOF Character 0
 Error Replace Character 0
 Error Replacement Enabled No
 Event Character 0
 Parity Check Enabled No
 RTS Flow Control Type Enable
 XOff Character 19
 XOffXMit Threshold 512
 XOn Character 17
 XOnXMit Threshold 2048
 XOnXOff InFlow Control 0
 XOnXOff OutFlow Control 0
 I/O Port 0x000002F8-0x000002FF
 IRQ Channel IRQ 3
 Driver d:\windows\system32\drivers\serial.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 64.00
 KB (65,536 bytes), 3/25/2003 8:00 PM)

[Parallel]

Item Value

[Storage]

[Drives]

Item Value

Drive B:
 Description Network Connection
 Provider Name \\192.168.0.161\mykit

Drive C:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 24.41 GB (26,213,933,056 bytes)
 Free Space 3.31 GB (3,552,993,280 bytes)
 Volume Name
 Volume Serial Number 98D2FFC5

Drive D:
 Description Local Fixed Disk
 Compressed No
 File System NTFS
 Size 24.41 GB (26,213,933,056 bytes)
 Free Space 18.71 GB (20,084,932,608 bytes)
 Volume Name

Volume Serial Number 74FC1B63

Drive E:

Description Local Fixed Disk
Compressed No
File System NTFS
Size 25.69 GB (27,587,555,328 bytes)
Free Space 12.61 GB (13,537,615,872 bytes)
Volume Name
Volume Serial Number 1C8833B3

Drive F:

Description CD-ROM Disc

[Disks]

Item Value
Description Disk drive
Manufacturer (Standard disk drives)
Model ST380013AS
Bytes/Sector 512
Media Loaded Yes
Media Type Fixed hard disk
Partitions 3
SCSI Bus 0
SCSI Logical Unit 0
SCSI Port 2
SCSI Target ID 0
Sectors/Track 63
Size 74.53 GB (80,023,749,120 bytes)
Total Cylinders 9,729
Total Sectors 156,296,385
Total Tracks 2,480,895
Tracks/Cylinder 255
Partition Disk #0, Partition #0
Partition Size 24.41 GB (26,213,935,104 bytes)
Partition Starting Offset 32,256 bytes
Partition Disk #0, Partition #1
Partition Size 50.11 GB (53,801,556,480 bytes)
Partition Starting Offset 26,213,967,360 bytes

[SCSI]

Item Value

[IDE]

Item Value
Name Intel(R) 82801EB Ultra ATA Storage Controllers - 24DB
Manufacturer Intel
Status OK
PNP Device ID
PCI\VEN_8086&DEV_24DB&SUBSYS_10798086&REV_02\3&267A616A&0&F9

I/O Port 0x0000FC00-0x0000FC0F
Memory Address 0xFFBFFC00-0xFFBFFFFF
Driver d:\windows\system32\drivers\intelide.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 7.50 KB (7,680 bytes), 6/3/2005 10:24 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&1BDE9191&0&0
I/O Port 0x000001F0-0x000001F7
I/O Port 0x000003F6-0x000003F6
IRQ Channel IRQ 14
Driver d:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/25/2003 8:00 PM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&1BDE9191&0&1
I/O Port 0x00000170-0x00000177
I/O Port 0x00000376-0x00000376
IRQ Channel IRQ 15
Driver d:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/25/2003 8:00 PM)

Name Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1
Manufacturer Intel
Status OK
PNP Device ID PCI\VEN_8086&DEV_24D1&SUBSYS_34378086&REV_02\3&267A616A&0&FA
I/O Port 0x0000C480-0x0000C487
I/O Port 0x0000C400-0x0000C403
I/O Port 0x0000C080-0x0000C087
I/O Port 0x0000C000-0x0000C003
I/O Port 0x0000BC00-0x0000BC0F
IRQ Channel IRQ 18
Driver d:\windows\system32\drivers\intelide.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 7.50 KB (7,680 bytes), 6/3/2005 10:24 PM)

Name Primary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&2AB0FAC9&0&0
Driver d:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/25/2003 8:00 PM)

Name Secondary IDE Channel
Manufacturer (Standard IDE ATA/ATAPI controllers)
Status OK
PNP Device ID PCIIDE\IDECHANNEL\4&2AB0FAC9&0&1
Driver d:\windows\system32\drivers\atapi.sys (5.2.3790.1830 (srv03_sp1_rtm.050324-1447), 93.50 KB (95,744 bytes), 3/25/2003 8:00 PM)

[Printing]

Name	Driver	Port Name	Server Name
------	--------	-----------	-------------

[Problem Devices]

Device	PNP Device ID	Error Code
--------	---------------	------------

[USB]

Device	PNP Device ID
--------	---------------

Standard	Universal	PCI	to	USB	Host	Controller
	PCIIVEN_8086&DEV_24D2&SUBSYS_10798086&REV_02\3&267A616A&0&E8					
Standard	Universal	PCI	to	USB	Host	Controller
	PCIIVEN_8086&DEV_24D4&SUBSYS_10798086&REV_02\3&267A616A&0&E9					
Standard	Universal	PCI	to	USB	Host	Controller
	PCIIVEN_8086&DEV_24D7&SUBSYS_10798086&REV_02\3&267A616A&0&EA					
Standard	Enhanced	PCI	to	USB	Host	Controller
	PCIIVEN_8086&DEV_24DD&SUBSYS_10798086&REV_02\3&267A616A&0&EF					

[Software Environment]

[System Drivers]

Name	Description	File	Type	Started	Start Mode	State	Status	Error	Control
	Accept Pause	Accept	Stop						
abiosdsk	Abiosdsk	Not Available	Kernel Driver	No	Disabled	Stopped			
	OK	Ignore	No	No					
acpi	Microsoft ACPI Driver	d:\windows\system32\drivers\acpi.sys	Kernel Driver	Yes					
	Boot	Running	OK	Normal	No	Yes			
acpiec	ACPIEC	d:\windows\system32\drivers\acpiec.sys	Kernel Driver	No					
	Disabled	Stopped	OK	Normal	No	No			
adpu160m	adpu160m	Not Available	Kernel Driver	No	Disabled	Stopped			
	OK	Normal	No	No					
adpu320	adpu320	Not Available	Kernel Driver	No	Disabled	Stopped			
	OK	Normal	No	No					
afcnt	afcnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK		Normal
	No	No							
afd	AFD Networking Support Environment	d:\windows\system32\drivers\afd.sys	Kernel						
Driver	Yes	System Running	OK	Normal	No	Yes			
aha154x	Aha154x	Not Available	Kernel Driver	No	Disabled	Stopped	OK		
	Normal	No	No						
aic78u2	aic78u2	Not Available	Kernel Driver	No	Disabled	Stopped	OK		Normal
	No	No							
aic78xx	aic78xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK		Normal
	No	No							
aliide	AliIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK		Normal
	No	No							
asyncmac	RAS Asynchronous Media Driver	d:\windows\system32\drivers\asyncmac.sys	Kernel Driver	No	Manual	Stopped	OK		Normal
	No	No							
atapi	Standard IDE/ESDI Hard Disk Controller	d:\windows\system32\drivers\atapi.sys	Kernel						

Driver	Yes	Boot	Running	OK	Normal	No	Yes			
atdisk	Atdisk	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Ignore		
	No	No								
ati2mpad		ati2mpad	d:\windows\system32\drivers\ati2mpad.sys	Kernel	Driver					
	Yes	Manual	Running	OK	Ignore	No	Yes			
atmarpc	ATM ARP Client Protocol		d:\windows\system32\drivers\atmarpc.sys	Kernel	Driver					
	No	Manual	Stopped	OK	Normal	No	No			
audstub	Audio Stub Driver		d:\windows\system32\drivers\audstub.sys	Kernel	Driver	Yes				
	Manual	Running	OK	Normal	No	Yes				
beep	Beep	d:\windows\system32\drivers\beep.sys	Kernel	Driver	Yes	System				
	Running	OK	Normal	No	Yes					
cbidf2k	cbidf2k	d:\windows\system32\drivers\cbidf2k.sys	Kernel	Driver	No	Disabled				
	Stopped	OK	Normal	No	No					
cd20xrnt	cd20xrnt	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
cdfs	Cdfs	d:\windows\system32\drivers\cdfs.sys	File System	Driver	Yes					
	Disabled	Running	OK	Normal	No	Yes				
cdrom	CD-ROM Driver		d:\windows\system32\drivers\cdrom.sys	Kernel	Driver	Yes				
	System	Running	OK	Normal	No	Yes				
changer	Changer	Not Available	Kernel	Driver	No	System	Stopped	OK	Ignore	No
	No									
clusdisk	Cluster Disk Driver		d:\windows\system32\drivers\clusdisk.sys	Kernel	Driver	No				
	Disabled	Stopped	OK	Normal	No	No				
cmdide	CmdIde	Not Available	Kernel	Driver	No	Disabled	Stopped	OK	Normal	
	No	No								
cpqarray	Cpqarray	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
cpqarry2	cpqarry2	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
cpqcissm	cpqcissm	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
cpqfcalm	cpqfcalm	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
credisk	CRC Disk Filter Driver		d:\windows\system32\drivers\credisk.sys	Kernel	Driver	Yes				
	Boot	Running	OK	Normal	No	Yes				
dac960nt	dac960nt	Not Available	Kernel	Driver	No	Disabled	Stopped			
	OK	Normal	No	No						
dellcerc	dellcerc	Not Available	Kernel	Driver	No	Disabled	Stopped	OK	Normal	
	No	No								
dfsdriver	DfsDriver		d:\windows\system32\drivers\dfs.sys	File	System	Driver				
	Yes	Boot	Running	OK	Normal	No	Yes			
disk	Disk Driver		d:\windows\system32\drivers\disk.sys	Kernel	Driver	Yes	Boot			
	Running	OK	Normal	No	Yes					
dmboot	dmboot	d:\windows\system32\drivers\dmboot.sys	Kernel	Driver	No	Disabled				
	Stopped	OK	Normal	No	No					
dmio	Logical Disk Manager Driver		d:\windows\system32\drivers\dmio.sys	Kernel	Driver					
	Yes	Boot	Running	OK	Normal	No	Yes			
dmload	dmload	d:\windows\system32\drivers\dmload.sys	Kernel	Driver	Yes	Boot				
	Running	OK	Normal	No	Yes					
dpti2o	dpti2o	Not Available	Kernel	Driver	No	Disabled	Stopped	OK	Normal	
	No	No								
e1000	Intel(R) PRO/1000 Adapter Driver		d:\windows\system32\drivers\e1000325.sys							
	Kernel	Driver	Yes	Manual	Running	OK	Normal	No	Yes	

fastfat	Fastfat	d:\windows\system32\drivers\fastfat.sys	File System Driver	No				
	Disabled	Stopped OK	Normal	No	No			
fdc	Fdc	d:\windows\system32\drivers\fdc.sys	Kernel Driver	No	System	Stopped		
	OK	Ignore	No	No				
fips	Fips	d:\windows\system32\drivers\fips.sys	Kernel Driver	Yes	System			
	Running	OK	Normal	No	Yes			
flpydisk	Flpydisk	d:\windows\system32\drivers\flpydisk.sys	Kernel Driver	No	System			
	Stopped OK	Ignore	No	No				
fltmgr	FltMgr	d:\windows\system32\drivers\fltmgr.sys	File System Driver	Yes	Boot			
	Running	OK	Normal	No	Yes			
ftdisk	Volume Manager Driver	d:\windows\system32\drivers\ftdisk.sys	Kernel Driver	Yes				
	Boot	Running	OK	Normal	No	Yes		
gpc	Generic Packet Classifier	d:\windows\system32\drivers\msgpc.sys	Kernel Driver	Yes				
	Manual	Running	OK	Normal	No	Yes		
hpn	hpn	Not Available	Kernel Driver	No	Disabled	Stopped OK	Normal	
	No	No						
hpt3xx	hpt3xx	Not Available	Kernel Driver	No	Disabled	Stopped OK	Normal	
	No	No						
http	HTTP	d:\windows\system32\drivers\http.sys	Kernel Driver	Yes	Manual			
	Running	OK	Normal	No	Yes			
i2omgmt	i2omgmt	Not Available	Kernel Driver	No	System	Stopped OK		
	Normal	No	No					
i2omp	i2omp	Not Available	Kernel Driver	No	Disabled	Stopped OK	Normal	
	No	No						
i8042prt	i8042 Keyboard and PS/2 Mouse Port Driver	d:\windows\system32\drivers\i8042prt.sys	Kernel Driver	Yes	System	Running	OK	Normal
	Kernel Driver	Yes	System	Running	OK	Normal	No	Yes
iirsp	iirsp	Not Available	Kernel Driver	No	Disabled	Stopped OK	Normal	
	No	No						
imapi	CD-Burning Filter Driver	d:\windows\system32\drivers\imapi.sys	Kernel Driver	No				
	System	Stopped OK	Normal	No	No			
intelide	IntelIde	d:\windows\system32\drivers\intelide.sys	Kernel Driver	Yes	Boot			
	Running	OK	Normal	No	Yes			
intelppm	Intel Processor Driver	d:\windows\system32\drivers\intelppm.sys	Kernel Driver	Yes	Manual	Running	OK	Normal
	Yes	Manual	Running	OK	Normal	No	Yes	
ip6fw	IPv6 Windows Firewall Driver	d:\windows\system32\drivers\ip6fw.sys	Kernel Driver	No	Manual	Stopped OK	Normal	No
	No	Manual	Stopped OK	Normal	No	No		
ipfilterdriver	IP Traffic Filter Driver	d:\windows\system32\drivers\ipfltdrv.sys	Kernel Driver	No	Manual	Stopped OK	Normal	No
	No	Manual	Stopped OK	Normal	No	No		
ipinip	IP in IP Tunnel Driver	d:\windows\system32\drivers\ipinip.sys	Kernel Driver	No	Manual	Stopped OK	Normal	No
	Manual	Stopped OK	Normal	No	No			
ipnat	IP Network Address Translator	d:\windows\system32\drivers\ipnat.sys	Kernel Driver	No	Manual	Stopped OK	Normal	No
	No	Manual	Stopped OK	Normal	No	No		
ipsec	IPSEC driver	d:\windows\system32\drivers\ipsec.sys	Kernel Driver	Yes	System			
	Running	OK	Normal	No	Yes			
ipsraidn	ipsraidn	Not Available	Kernel Driver	No	Disabled	Stopped OK	Normal	
	No	No						
irenum	IR Enumerator Service	d:\windows\system32\drivers\irenum.sys	Kernel Driver	No	Manual	Stopped OK	Normal	No
	Manual	Stopped OK	Normal	No	No			
isapnp	PnP ISA/EISA Bus Driver	d:\windows\system32\drivers\isapnp.sys	Kernel Driver	Yes	Boot	Running	OK	Critical
	Yes	Boot	Running	OK	Critical	No	Yes	
kbdclass	Keyboard Class Driver	d:\windows\system32\drivers\kbdclass.sys	Kernel Driver	Yes	System	Running	OK	Normal
	Yes	System	Running	OK	Normal	No	Yes	
ksecdd	KSecDD	d:\windows\system32\drivers\ksecdd.sys	Kernel Driver	Yes	Boot			

	Running	OK	Normal	No	Yes			
lp6nds35	lp6nds35	Not Available	Kernel Driver	No	Disabled	Stopped		
	OK	Normal	No	No				
mnmdd	mnmdd	d:\windows\system32\drivers\mnmdd.sys	Kernel Driver	Yes	System			
	Running	OK	Ignore	No	Yes			
modem	Modem	d:\windows\system32\drivers\modem.sys	Kernel Driver	No	Manual	Stopped		
	OK	Ignore	No	No				
mouclass	Mouse Class Driver	d:\windows\system32\drivers\mouclass.sys	Kernel					
Driver	Yes	System	Running	OK	Normal	No	Yes	
mountmgr	Mount Point Manager	d:\windows\system32\drivers\mountmgr.sys	Kernel					
Driver	Yes	Boot	Running	OK	Normal	No	Yes	
mraid35x	mraid35x	Not Available	Kernel Driver	No	Disabled	Stopped		
	OK	Normal	No	No				
mrxdav	WebDav Client Redirector	d:\windows\system32\drivers\mrxdav.sys	File		System			
Driver	No	Manual	Stopped	OK	Normal	No	No	
mrxsmb	MRXSMB	d:\windows\system32\drivers\mrxsmb.sys	File System Driver		Yes			
	System	Running	OK	Normal	No	Yes		
msfs	Msfs	d:\windows\system32\drivers\msfs.sys	File System Driver	Yes	System			
	Running	OK	Normal	No	Yes			
mssmbios	Microsoft	System	Management	BIOS	Driver			
	d:\windows\system32\drivers\mssmbios.sys	Kernel Driver	Yes	Manual				
	Running	OK	Normal	No	Yes			
mup	Mup	d:\windows\system32\drivers\mup.sys	File System Driver	Yes	Boot			
	Running	OK	Normal	No	Yes			
ndis	NDIS System Driver	d:\windows\system32\drivers\ndis.sys	Kernel Driver	Yes				
	Boot	Running	OK	Normal	No	Yes		
ndistapi	Remote Access NDIS TAPI Driver	d:\windows\system32\drivers\ndistapi.sys	Kernel					
Driver	Yes	Manual	Running	OK	Normal	No	Yes	
ndisuio	NDIS Usermode I/O Protocol	d:\windows\system32\drivers\ndisuio.sys	Kernel	Driver				
	Yes	Manual	Running	OK	Normal	No	Yes	
ndiswan	Remote Access NDIS WAN Driver	d:\windows\system32\drivers\ndiswan.sys	Kernel					
Driver	Yes	Manual	Running	OK	Normal	No	Yes	
ndproxy	NDIS Proxy	d:\windows\system32\drivers\ndproxy.sys	Kernel Driver	Yes	Manual			
	Running	OK	Normal	No	Yes			
netbios	NetBIOS Interface	d:\windows\system32\drivers\netbios.sys	File	System	Driver			
	Yes	System	Running	OK	Normal	No	Yes	
netbt	NetBios over Tcpip	d:\windows\system32\drivers\netbt.sys	Kernel Driver	Yes				
	System	Running	OK	Normal	No	Yes		
nfrd960	nfrd960	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No						
npfs	Npfs	d:\windows\system32\drivers\npfs.sys	File System Driver	Yes	System			
	Running	OK	Normal	No	Yes			
ntfs	Ntfs	d:\windows\system32\drivers\ntfs.sys	File System Driver	Yes				
	Disabled	Running	OK	Normal	No	Yes		
null	Null	d:\windows\system32\drivers\null.sys	Kernel Driver	Yes	System			
	Running	OK	Normal	No	Yes			
parport	Parport	d:\windows\system32\drivers\parport.sys	Kernel Driver	No	Manual	Stopped		
	OK	Ignore	No	No				
partmgr	Partition Manager	d:\windows\system32\drivers\partmgr.sys	Kernel Driver	Yes				
	Boot	Running	OK	Normal	No	Yes		
pci	PCI Bus Driver	d:\windows\system32\drivers\pci.sys	Kernel Driver	Yes	Boot			
	Running	OK	Critical	No	Yes			
pciide	PCIIde	d:\windows\system32\drivers\pciide.sys	Kernel Driver	Yes	Boot			

	Running	OK	Normal	No	Yes			
pcmcia	Pcmcia d:\windows\system32\drivers\pcmcia.sys	Kernel Driver	No	Disabled				
	Stopped OK	Normal	No	No				
pdcomp	PDCOMP	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No No							
pdframe	PDFFRAME	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No No							
pdreli	PDRELI	Not Available	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No No							
pdrframe	PDRFRAME	Not Available	Kernel Driver	No	Manual	Stopped	OK	
	Ignore No No							
perc2	perc2	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
perc2hib	perc2hib	Not Available	Kernel Driver	No	Disabled	Stopped		
	OK Normal No No							
pptpminiport	WAN Miniport (PPTP) d:\windows\system32\drivers\raspptp.sys	Kernel Driver	No	Yes				
	Yes Manual Running OK Normal No Yes							
processor	Processor Driver d:\windows\system32\drivers\processr.sys	Kernel Driver	No					
	Manual Stopped OK Normal No No							
ptilink	Direct Parallel Link Driver d:\windows\system32\drivers\ptilink.sys	Kernel Driver	No	Yes				
	Yes Manual Running OK Normal No Yes							
ql1080	ql1080	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql10wnt	Ql10wnt	Not Available	Kernel Driver	No	Disabled	Stopped	OK	
	Normal No No							
ql12160	ql12160	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql1240	ql1240	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql1280	ql1280	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql2100	ql2100	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql2200	ql2200	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
ql2300	ql2300	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No No							
rasacd	Remote Access Auto Connection Driver d:\windows\system32\drivers\rasacd.sys	Kernel Driver	No	Yes				
Driver	Yes System Running OK Normal No Yes							
rasl2tp	WAN Miniport (L2TP) d:\windows\system32\drivers\rasl2tp.sys	Kernel Driver	No	Yes				
	Manual Running OK Normal No Yes							
rasppoe	Remote Access PPPOE Driver d:\windows\system32\drivers\rasppoe.sys	Kernel Driver	Yes	Manual Running OK Normal No Yes				
	Kernel Driver Yes Manual Running OK Normal No Yes							
raspti	Direct Parallel d:\windows\system32\drivers\raspti.sys	Kernel Driver	Yes	Manual Running OK Normal No Yes				
	Running OK Normal No Yes							
rdcss	Rdbss d:\windows\system32\drivers\rdbss.sys	File System Driver	Yes	System Running OK Normal No Yes				
	Running OK Normal No Yes							
rdpcdd	RDPcDD d:\windows\system32\drivers\rdpcdd.sys	Kernel Driver	Yes	System Running OK Ignore No Yes				
	Running OK Ignore No Yes							
rdpdr	Terminal Server Device Redirector Driver d:\windows\system32\drivers\rdpdr.sys	Kernel Driver	No	Yes				
Driver	Yes Manual Running OK Normal No Yes							
rdpwd	RDPWD d:\windows\system32\drivers\rdpwd.sys	Kernel Driver	Yes	Manual Running OK Ignore No Yes				
	Running OK Ignore No Yes							

redbook	Digital CD Audio Playback Filter Driver	d:\windows\system32\drivers\redbook.sys	Kernel Driver	Yes	System Running	OK	Normal	No	Yes		
secdrv	Secdrv	d:\windows\system32\drivers\secdrv.sys	Kernel Driver	No	Manual Stopped	OK	Normal	No	No		
serenum	Serenum Filter Driver	d:\windows\system32\drivers\serenum.sys	Kernel Driver	Yes		Manual Running	OK	Normal	No	Yes	
serial	Serial port driver	d:\windows\system32\drivers\serial.sys	Kernel Driver	Yes	System	Running	OK	Ignore	No	Yes	
sfloppy	Sfloppy	d:\windows\system32\drivers\sfloppy.sys	Kernel Driver	No	System Stopped	OK	Ignore	No	No		
simbad	Simbad	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
sparrow	Sparrow	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
srv	Srv	d:\windows\system32\drivers\srv.sys	File System Driver	Yes	Manual	Running	OK	Normal	No	Yes	
swenum	Software Bus Driver	d:\windows\system32\drivers\swenum.sys	Kernel Driver	Yes		Manual Running	OK	Normal	No	Yes	
symc810	symc810	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
symc8xx	symc8xx	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
symmpi	symmpi	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
sym_hi	sym_hi	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
sym_u3	sym_u3	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
tcpip	TCP/IP Protocol Driver	d:\windows\system32\drivers\tcpip.sys	Kernel Driver	Yes		System Running	OK	Normal	No	Yes	
tdpipe	TDPIPE	d:\windows\system32\drivers\tdpipe.sys	Kernel Driver	No	Manual Stopped	OK	Ignore	No	No		
tdtcp	TDTCP	d:\windows\system32\drivers\tdtcp.sys	Kernel Driver	Yes	Manual	Running	OK	Ignore	No	Yes	
termdd	Terminal Device Driver	d:\windows\system32\drivers\termdd.sys	Kernel Driver	Yes		System Running	OK	Normal	No	Yes	
toside	TosIde	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
udfs	Udfs	d:\windows\system32\drivers\udfs.sys	File System Driver	No		Disabled	Stopped	OK	Normal	No	
ultra	ultra	Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal	No	No	
update	Microcode Update Driver	d:\windows\system32\drivers\update.sys	Kernel Driver	Yes		Manual Running	OK	Normal	No	Yes	
usbehci	Microsoft USB 2.0 Enhanced Host Controller	d:\windows\system32\drivers\usbehci.sys	Kernel Driver	Yes	Miniport Driver	Manual Running	OK	Normal	No	Yes	
usbhub	USB2 Enabled Hub	d:\windows\system32\drivers\usbhub.sys	Kernel Driver	Yes		Manual Running	OK	Normal	No	Yes	
usbuhci	Microsoft USB Universal Host Controller	d:\windows\system32\drivers\usbuhci.sys	Kernel Driver	Yes	Miniport Driver	Manual Running	OK	Normal	No	Yes	
vgasave	VGA Display Controller	d:\windows\system32\drivers\vga.sys	Kernel Driver	Yes							

	System Running	OK	Ignore	No	Yes		
viaide	ViaIde Not Available	Kernel Driver	No	Disabled	Stopped	OK	Normal
	No	No					
volsnap	Storage volumes d:\windows\system32\drivers\volsnap.sys	Kernel Driver	Yes	Boot			
	Running	OK	Normal	No	Yes		
wanarp	Remote Access IP ARP Driver	d:\windows\system32\drivers\wanarp.sys	Kernel	Driver			
	Yes	Manual Running	OK	Normal	No	Yes	
wdica	WDICANot Available	Kernel Driver	No	Manual	Stopped	OK	Ignore
	No						No
wlbs	Network Load Balancing d:\windows\system32\drivers\wlbs.sys	Kernel Driver	No				
	Manual	Stopped	OK	Normal	No	No	

[Signed Drivers]

Device Name	Signed	Device Class	Driver Version	Driver Date	Manufacturer	INF
Name	Driver Name	Device ID				
Microsoft System Management BIOS	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
10/1/2002	(Standard system devices)	machine.inf	Not Available	ROOT\SYSTEM\0002		Available
Microcode Update Device	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
(Standard system devices)	machine.inf	Not Available	ROOT\SYSTEM\0001			
Plug and Play Software Device Enumerator	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
10/1/2002	(Standard system devices)	machine.inf	Not Available	ROOT\SYSTEM\0000		Available
Terminal Server Mouse Driver	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
(Standard system devices)	machine.inf	Not Available	ROOT\RDP_MOU\0000			
Terminal Server Keyboard Driver	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
(Standard system devices)	machine.inf	Not Available	ROOT\RDP_KBD\0000			
Terminal Server Device Redirector	Yes	SYSTEM	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
(Standard system devices)	machine.inf	Not Available	ROOT\RDPRD\0000			
Direct Parallel	Yes	NET	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
Not Available	ROOT\MS_PTMINIPORT\0000					
WAN Miniport (PPTP)	Yes	NET	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
netrasa.inf	Not Available	ROOT\MS_PPTPMINIPORT\0000				
WAN Miniport (PPPOE)	Yes	NET	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
netrasa.inf	Not Available	ROOT\MS_PPPOEMINIPORT\0000				
WAN Miniport (IP)	Yes	NET	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
netrasa.inf	Not Available	ROOT\MS_NDISWANIP\0000				
WAN Miniport (L2TP)	Yes	NET	5.2.3790.0	10/1/2002	Microsoft	netrasa.inf
netrasa.inf	Not Available	ROOT\MS_L2TPMINIPORT\0000				
Video Codecs	Yes	MEDIA	5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf	Not Available	ROOT\MEDIA\MS_MMVID				
Legacy Video Capture Devices	Yes	MEDIA	5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf	Not Available	ROOT\MEDIA\MS_MMVCD				
Media Control Devices	Yes	MEDIA	5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf	Not Available	ROOT\MEDIA\MS_MMMCI				
Legacy Audio Drivers	Yes	MEDIA	5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf	Not Available	ROOT\MEDIA\MS_MMDRV				
Audio Codecs	Yes	MEDIA	5.2.3790.0	10/1/2002	(Standard system devices)	
wave.inf	Not Available	ROOT\MEDIA\MS_MMACM				
Remote Access IP ARP Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not Available	Not Available
Available	Not Available	Not Available	Not Available	Not Available	Not Available	Available
		ROOT\LEGACY_WANARP\0000				

volsnap	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_VOLSNAP\0000			
VGA Display Controller.	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_VGASAVE\0000		
TDTCP	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_TDTCP\0000			
TCP/IP Protocol Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_TCPIP\0000		
RDPWD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOTLEGACY_RDPWD\0000			
RDPCDD	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOTLEGACY_RDPCDD\0000			
Remote Access Auto Connection Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	Not		Available
			ROOTLEGACY_RASACD\0000			
PCIIde	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_PCIIDE\0000			
Partition Manager	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_PARTMGR\0000		
Null	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_NULL\0000			
NetBios over Tcpi	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_NETBT\0000		
NDProxy	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOTLEGACY_NDPROXY\0000			
NDIS Usermode I/O Protocol	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	Not			Available
			ROOTLEGACY_NDISUIO\0000			
Remote Access NDIS TAPI Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	Not		Available
			ROOTLEGACY_NDISTAPI\0000			
NDIS System Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_NDIS\0000		
mountmgr	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOTLEGACY_MOUNTMGR\0000			
mmdd	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_MNMDD\0000			
ksecdd	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_KSECDD\0000			
IPSEC driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	
Available	Not Available	Not Available	ROOTLEGACY_IPSEC\0000			
HTTP	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_HTTP\0000			
Generic Packet Classifier	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_GPC\0000		
Fips	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_FIPS\0000			
dmload	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_DMLOAD\0000			
dmboot	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_DMBOOT\0000			
CRC Disk Filter Driver	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	Not Available	ROOTLEGACY_CRCDISK\0000		

Beep	Not Available	LEGACYDRIVER	Not Available	Not Available	Not	Available
	Not Available	Not Available	ROOTLEGACY_BEEP\0000			
AFD Networking Support Environment	Not Available	LEGACYDRIVER	Not Available	LEGACYDRIVER	Not	Available
	Not Available	Not Available	Not Available	Not		Available
		ROOTLEGACY_AFD\0000				
Generic volume	Yes	VOLUME	5.2.3790.0	10/1/2002	Microsoft	
volume.inf	Not	Not				Available
		STORAGE\VOLUME\1&30A96598&0&SIGNATURE128F128EOFFSETC34F34A00LEN				
GTH66C58B600						
Generic volume	Yes	VOLUME	5.2.3790.0	10/1/2002	Microsoft	
volume.inf	Not	Not				Available
		STORAGE\VOLUME\1&30A96598&0&SIGNATURE128F128EOFFSET61A79E400LEN				
GTH61A78E800						
Generic volume	Yes	VOLUME	5.2.3790.0	10/1/2002	Microsoft	
volume.inf	Not	Not				Available
		STORAGE\VOLUME\1&30A96598&0&SIGNATURE128F128EOFFSET7E00LENGTH61				
A78E800						
Volume Manager	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard	
system devices) machine.inf	Not Available	Not Available	ROOTFTDISK\0000			
Logical Disk Manager	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard	
system devices) machine.inf	Not Available	Not Available	ROOTDMIO\0000			
ACPI Fixed Feature Button	Yes	SYSTEM	5.2.3790.0	10/1/2002		
(Standard system devices)		machine.inf	Not			Available
		ACPI\FIXEDBUTTON\2&DABA3FF&0				
ACPI Power Button	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard	
system devices) machine.inf	Not Available	Not Available	ACPI\PNP0C0C\AA			
System board	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)	
machine.inf	Not Available	Not Available	ACPI\PNP0C01\1			
Motherboard resources	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard	
system devices) machine.inf	Not Available	Not Available	ACPI\PNP0C02\11			
Intel(R) 82801EB SMBus Controller - 24D3	Yes	SYSTEM	5.2.3790.1830			
10/1/2002	Intel	machine.inf	Not			Available
		PCI\VEN_8086&DEV_24D3&SUBSYS_10798086&REV_02\3&267A616A&0&FB				
Disk drive	Yes	DISKDRIVE	5.2.3790.0	10/1/2002	(Standard disk drives)	
disk.inf	Not	Not				Available
		IDE\DISKST380013AS_____3.05____\4A3545564B3953582				
020202020202020202020202020202020						
Secondary IDE Channel	Yes	HDC	5.2.3790.0	10/1/2002	(Standard	IDE
ATA/ATAPI controllers) mshdc.inf	Not	Not				Available
		PCIIDE\IDECHANNEL\4&2AB0FAC9&0&1				
Primary IDE Channel	Yes	HDC	5.2.3790.0	10/1/2002	(Standard	IDE
ATA/ATAPI controllers) mshdc.inf	Not	Not				Available
		PCIIDE\IDECHANNEL\4&2AB0FAC9&0&0				
Intel(R) 82801EB Ultra ATA Storage Controllers - 24D1	Yes	HDC	5.2.3790.1830			
10/1/2002	Intel	mshdc.inf	Not			Available
		PCI\VEN_8086&DEV_24D1&SUBSYS_34378086&REV_02\3&267A616A&0&FA				
CD-ROM Drive	Yes	CDROM	5.2.3790.0	10/1/2002	(Standard	CD-ROM
drives) cdrom.inf	Not Available	Not Available	IDE\CDROMMITSUMI_CD-			
		ROM_SR244W_____T01A____\5&20AE57A1&0&0.1.0				
Secondary IDE Channel	Yes	HDC	5.2.3790.0	10/1/2002	(Standard	IDE
ATA/ATAPI controllers) mshdc.inf	Not	Not				Available
		PCIIDE\IDECHANNEL\4&1BDE9191&0&1				
Primary IDE Channel	Yes	HDC	5.2.3790.0	10/1/2002	(Standard	IDE

ATA/ATAPI controllers)	mshdc.inf	Not			Available
	PCI\IDE\IDECHANNEL\4&1BDE9191&0&0				
Intel(R) 82801EB Ultra ATA Storage Controllers - 24DB	Yes	HDC	5.2.3790.1830		
10/1/2002	Intel	mshdc.inf	Not		Available
	PCI\VEN_8086&DEV_24DB&SUBSYS_10798086&REV_02\3&267A616A&0&F9				
Motherboard resources	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard
system devices) machine.inf	Not Available	ACPI\PNP0C02\0			
Motherboard resources	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard
system devices) machine.inf	Not Available	ACPI\PNP0C02\10			
Motherboard resources	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard
system devices) machine.inf	Not Available	ACPI\PNP0C02\2E			
Communications Port	Yes	PORTS	5.2.3790.0	10/1/2002	(Standard port types)
msports.inf	Not Available	ACPI\PNP0501\2			
Communications Port	Yes	PORTS	5.2.3790.0	10/1/2002	(Standard port types)
msports.inf	Not Available	ACPI\PNP0501\1			
Numeric data processor	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard
system devices) machine.inf	Not Available	ACPI\PNP0C04\4&35F762C4&0			
System speaker	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\PNP0800\4&35F762C4&0			
Microsoft PS/2 Mouse	Yes	MOUSE	5.2.3790.0	10/1/2002	Microsoft
msmouse.inf	Not Available	ACPI\PNP0F03\4&35F762C4&0			
Standard 101/102-Key or Microsoft Natural PS/2 Keyboard	Yes	KEYBOARD	5.2.3790.0		
10/1/2002	(Standard keyboards)	keyboard.inf	Not		Available
	ACPI\PNP0303\4&35F762C4&0				
System CMOS/real time clock	Yes	SYSTEM	5.2.3790.0	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
	ACPI\PNP0B00\4&35F762C4&0				
System timer	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)
machine.inf	Not Available	ACPI\PNP0100\4&35F762C4&0			
Direct memory access controller	Yes	SYSTEM	5.2.3790.0	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
	ACPI\PNP0200\4&35F762C4&0				
Programmable interrupt controller	Yes	SYSTEM	5.2.3790.0	10/1/2002	
(Standard system devices)		machine.inf	Not		Available
	ACPI\PNP0000\4&35F762C4&0				
ISAPNP Read Data Port	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard
system devices) machine.inf	Not Available	ISAPNP\READDATAPORT\0			
Intel(R) 82801EB LPC Interface Controller - 24D0	Yes	SYSTEM	5.2.3790.1830		
10/1/2002	Intel	machine.inf	Not		Available
	PCI\VEN_8086&DEV_24D0&SUBSYS_00000000&REV_02\3&267A616A&0&F8				
Default Monitor	Yes	MONITOR	5.1.2001.0	6/6/2001	(Standard monitor types)
monitor.inf	Not				Available
	DISPLAY\DEFAULT_MONITOR\5&1C603F11&0&80000000&04&0C				
Default Monitor	Yes	MONITOR	5.1.2001.0	6/6/2001	(Standard monitor types)
monitor.inf	Not				Available
	DISPLAY\DEFAULT_MONITOR\5&1C603F11&0&80000000A&04&0C				
RAGE XL PCI Family (Microsoft Corporation)	Yes	DISPLAY	5.10.2600.6014		
8/8/2001	ATI Technologies Inc.	atiixpad.inf	Not		Available
	PCI\VEN_1002&DEV_4752&SUBSYS_10798086&REV_27\4&2E98101C&0&60F0				
Intel(R) 82801 PCI Bridge - 244E	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel
machine.inf	Not				Available
	PCI\VEN_8086&DEV_244E&SUBSYS_00000000&REV_C2\3&267A616A&0&F0				
USB Root Hub	Yes	USB	5.2.3790.0	10/1/2002	(Standard USB Host Controller)

	usbport.inf	Not Available	USB\ROOT_HUB20\4&3176BA1E&0			
Standard Enhanced PCI to USB Host Controller	Yes	USB	5.2.3790.0	10/1/2002		
(Standard USB Host Controller)	usbport.inf	Not Available				Available
	PCI\VEN_8086&DEV_24DD&SUBSYS_10798086&REV_02\3&267A616A&0&EF					
USB Root Hub	Yes	USB	5.2.3790.0	10/1/2002	(Standard USB Host Controller)	
	usbport.inf	Not Available	USB\ROOT_HUB\4&AA4272E&0			
Standard Universal PCI to USB Host Controller	Yes	USB	5.2.3790.0	10/1/2002		
(Standard USB Host Controller)	usbport.inf	Not Available				Available
	PCI\VEN_8086&DEV_24D7&SUBSYS_10798086&REV_02\3&267A616A&0&EA					
USB Root Hub	Yes	USB	5.2.3790.0	10/1/2002	(Standard USB Host Controller)	
	usbport.inf	Not Available	USB\ROOT_HUB\4&2443A96&0			
Standard Universal PCI to USB Host Controller	Yes	USB	5.2.3790.0	10/1/2002		
(Standard USB Host Controller)	usbport.inf	Not Available				Available
	PCI\VEN_8086&DEV_24D4&SUBSYS_10798086&REV_02\3&267A616A&0&E9					
USB Root Hub	Yes	USB	5.2.3790.0	10/1/2002	(Standard USB Host Controller)	
	usbport.inf	Not Available	USB\ROOT_HUB\4&33DAAAF6&0			
Standard Universal PCI to USB Host Controller	Yes	USB	5.2.3790.0	10/1/2002		
(Standard USB Host Controller)	usbport.inf	Not Available				Available
	PCI\VEN_8086&DEV_24D2&SUBSYS_10798086&REV_02\3&267A616A&0&E8					
Intel(R) 6700PXH I/OxAPIC Interrupt Controller B - 0327	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_0327&SUBSYS_10798086&REV_09\4&117DD1A8&0&0310					Available
Intel(R) PRO/1000 MT Dual Port Network Connection	Yes	NET	7.4.19.0 3/12/2004			
	Intel oem2.inf	Not Available				Available
	PCI\VEN_8086&DEV_1079&SUBSYS_10798086&REV_03\5&CBC31F2&0&210210					
Intel(R) PRO/1000 MT Dual Port Network Connection	Yes	NET	7.4.19.0 3/12/2004			
	Intel oem2.inf	Not Available				Available
	PCI\VEN_8086&DEV_1079&SUBSYS_10798086&REV_03\5&CBC31F2&0&200210					
Intel(R) 6700PXH PCI Express-to-PCI Bridge B - 032A	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_032A&SUBSYS_00000000&REV_09\4&117DD1A8&0&0210					Available
Intel(R) 6700PXH I/OxAPIC Interrupt Controller A - 0326	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_0326&SUBSYS_10798086&REV_09\4&117DD1A8&0&0110					Available
Intel(R) 6700PXH PCI Express-to-PCI Bridge A - 0329	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_0329&SUBSYS_00000000&REV_09\4&117DD1A8&0&0010					Available
Intel(R) E7525/E7520/E7320 PCI Express Root Port A0 - 3595	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_3595&SUBSYS_00000000&REV_0C\3&267A616A&0&10					Available
Intel(R) E7525/E7520 DMA Controller - 3594	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_3594&SUBSYS_10798086&REV_0C\3&267A616A&0&08					Available
Intel(R) E7525/E7520/E7320 Error Reporting Registers - 3591	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_3591&SUBSYS_10798086&REV_0C\3&267A616A&0&01					Available
Intel(R) E7520 Memory Controller Hub - 3590	Yes	SYSTEM	5.2.3790.1830	10/1/2002	Intel machine.inf	Not Available
	PCI\VEN_8086&DEV_3590&SUBSYS_00000000&REV_0C\3&267A616A&0&00					Available
PCI bus	Yes	SYSTEM	5.2.3790.0	10/1/2002	(Standard system devices)	
	machine.inf	Not Available	ACPI\PNP0A03\0			
Intel Processor	Yes	PROCESSOR	5.2.3790.1830	10/1/2002	Intel cpu.inf	Not Available
Available		ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_4\3				


```

Intel Processor Available Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_4\_2
Intel Processor Available Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_4\_1
Intel Processor Available Yes PROCESSOR 5.2.3790.1830 10/1/2002 Intel cpu.inf Not
ACPI\GENUINEINTEL_-X86_FAMILY_15_MODEL_4\_0
Microsoft ACPI-Compliant System Yes SYSTEM 5.2.3790.0 10/1/2002
Microsoft acpi.inf Not Available ACPI_HAL\PNP0C08\0
ACPI Multiprocessor PC Yes COMPUTER 5.2.3790.0 10/1/2002 (Standard
computers) hal.inf Not Available ROOT\ACPI_HAL\0000
Not Available Not Available Not Available Not Available Not Available Not Available
Not Available Not Available HTREE\ROOT\0

```

[Environment Variables]

```

Variable Value User Name
ComSpec %SystemRoot%\system32\cmd.exe <SYSTEM>
Path %SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;D:\Program
Files\Microsoft SQL Server\80\Tools\Binn\;D:\Program Files\Microsoft SQL
Server\90\Tools\Binn\;D:\Program Files\Microsoft SQL Server\90\DTS\Binn\;D:\Program
Files\Microsoft SQL Server\90\Tools\Binn\VSShell\Common7\IDE\;D:\Program Files\Microsoft
Visual Studio 8\Common7\IDE\PrivateAssemblies\ <SYSTEM>
windir %SystemRoot% <SYSTEM>
OS Windows_NT <SYSTEM>
PROCESSOR_ARCHITECTURE x86 <SYSTEM>
PROCESSOR_LEVEL 15 <SYSTEM>
PROCESSOR_IDENTIFIER x86 Family 15 Model 4 Stepping 1, GenuineIntel <SYSTEM>
PROCESSOR_REVISION 0401 <SYSTEM>
NUMBER_OF_PROCESSORS 4 <SYSTEM>
ClusterLog D:\WINDOWS\Cluster\cluster.log <SYSTEM>
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH <SYSTEM>
TEMP %SystemRoot%\TEMP <SYSTEM>
TMP %SystemRoot%\TEMP <SYSTEM>
FP_NO_HOST_CHECK NO <SYSTEM>
lib D:\Program Files\SQLXML 4.0\bin\ <SYSTEM>
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\SYSTEM
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\LOCAL SERVICE
TEMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE
TMP %USERPROFILE%\Local Settings\Temp NT AUTHORITY\NETWORK SERVICE
TEMP %USERPROFILE%\Local Settings\Temp WEB171\Administrator
TMP %USERPROFILE%\Local Settings\Temp WEB171\Administrator

```

[Print Jobs]

Document	Size	Owner	Notify	Status	Time Submitted	Start Time	Until	Time
	Elapsed Time	Pages Printed	Job ID	Priority	Parameters	Driver	Print	Processor
	Host Print Queue	Data Type	Name					

[Network Connections]

Local Name	Remote Name	Type	Status	User Name
------------	-------------	------	--------	-----------

B: \\192.168.0.161\mykit Disk Persistent Connection WEB171\Administrator

[Running Tasks]

Name	Path	Process ID	Priority	Min Working Set	Max Working Set	Start
system idle process		Not Available	0	0	Not Available	Not Available
Available		Not Available	Not Available	Not Available		
system	Not Available	4	8	0	1413120	Not Available
	Not Available	Not Available				Not Available
smss.exe		Not Available	584	11	204800 1413120	4/4/2006 11:18 AM
	Not Available	Not Available		Not Available		
csrss.exe		Not Available	696	13	Not Available	4/4/2006 11:18 AM
	Not Available	Not Available		Not Available		
winlogon.exe	d:\windows\system32\winlogon.exe			1312	13 204800 1413120	4/4/2006 11:19 AM
(508,928 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	497.00	KB
services.exe	d:\windows\system32\services.exe			1364	9 204800 1413120	4/4/2006 11:19 AM
(110,080 bytes)	3/25/2003 8:00 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	107.50	KB
lsass.exe	d:\windows\system32\lsass.exe			1384	9 204800 1413120	4/4/2006 11:19 AM
	3/25/2003 8:00 PM			5.2.3790.0 (srv03_rtm.030324-2048)	13.00	KB (13,312 bytes)
svchost.exe	d:\windows\system32\svchost.exe			1592	8 204800 1413120	4/4/2006 11:19 AM
(14,336 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB
svchost.exe	Not Available	1656	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
svchost.exe	Not Available	1748	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
svchost.exe	Not Available	1800	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
svchost.exe	d:\windows\system32\svchost.exe			1816	8 204800 1413120	4/4/2006 11:19 AM
(14,336 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB
spoolsv.exe	d:\windows\system32\spoolsv.exe			368	8 204800 1413120	4/4/2006 11:19 AM
(58,368 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	57.00	KB
msdtc.exe	Not Available	396	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
svchost.exe	d:\windows\system32\svchost.exe			644	8 204800 1413120	4/4/2006 11:19 AM
(14,336 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB
inetinfo.exe	d:\windows\system32\inetrv\inetinfo.exe			744	8 204800 1413120	4/4/2006 11:19 AM
(14,336 bytes)	11/22/2005 4:21 PM			6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB
svchost.exe	Not Available	796	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
rsys.exe	Not Available	888	8	Not Available	Not Available	4/4/2006 11:19 AM
	Not Available	Not Available		Not Available		
svchost.exe	d:\windows\system32\svchost.exe			1272	8 204800 1413120	4/4/2006 11:19 AM
(14,336 bytes)	11/22/2005 4:21 PM			5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB

```

svchost.exe    d:\windows\system32\svchost.exe 1788    8    204800 1413120
              4/4/2006 11:19 AM    5.2.3790.1830 (srv03_sp1_rtm.050324-1447)    14.00    KB
(14,336 bytes) 11/22/2005 4:21 PM
explorer.exe   d:\windows\explorer.exe 412    8    204800 1413120    4/4/2006 11:19
AM    6.00.3790.1830 (srv03_sp1_rtm.050324-1447)    1.00    MB    (1,050,624    bytes)
              11/22/2005 4:21 PM
cmd.exe        d:\windows\system32\cmd.exe 624    8    204800 1413120    4/4/2006 11:19
AM    5.2.3790.1830 (srv03_sp1_rtm.050324-1447)    379.00    KB    (388,096    bytes)
              3/25/2003 8:00 PM
wmiprvse.exe   Not Available 4092    8    Not Available    Not Available    4/4/2006 11:20
AM    Not Available    Not Available    Not Available
csrss.exe      Not Available 2396    13    Not Available    Not Available    4/4/2006 12:54
PM    Not Available    Not Available    Not Available
winlogon.exe   d:\windows\system32\winlogon.exe 2416    13    204800 1413120
              4/4/2006 12:54 PM    5.2.3790.1830 (srv03_sp1_rtm.050324-1447)    497.00    KB
(508,928 bytes) 11/22/2005 4:21 PM
rdpclip.exe    d:\windows\system32\rdpclip.exe 1996    8    204800 1413120
              4/4/2006 12:54 PM    5.2.3790.1830 (srv03_sp1_rtm.050324-1447)    68.00    KB
(69,632 bytes) 11/22/2005 4:21 PM
explorer.exe   d:\windows\explorer.exe 2200    8    204800 1413120    4/4/2006 12:54
PM    6.00.3790.1830 (srv03_sp1_rtm.050324-1447)    1.00    MB    (1,050,624    bytes)
              11/22/2005 4:21 PM
w3wp.exe       d:\windows\system32\inetsrv\w3wp.exe 1600    8    204800 1413120
              4/4/2006 12:56 PM    6.0.3790.1830 (srv03_sp1_rtm.050324-1447)    7.00    KB (7,168
bytes) 11/22/2005 4:21 PM
dllhost.exe    d:\windows\system32\dllhost.exe 3616    8    204800 1413120
              4/4/2006 12:56 PM    5.2.3790.0 (srv03_rtm.030324-2048)    5.50    KB (5,632    bytes)
              3/25/2003 8:00 PM
helpctr.exe    d:\windows\pchealth\helpctr\binaries\helpctr.exe 3420    8    204800
1413120    4/4/2006 9:05 PM    5.2.3790.1830    (srv03_sp1_rtm.050324-1447)
778.00 KB (796,672 bytes) 11/22/2005 4:21 PM
helpsvc.exe    d:\windows\pchealth\helpctr\binaries\helpsvc.exe 3676    8    204800
1413120    4/4/2006 9:05 PM    5.2.3790.1830    (srv03_sp1_rtm.050324-1447)
745.00 KB (762,880 bytes) 11/22/2005 4:21 PM
wmiprvse.exe   Not Available 1764    8    Not Available    Not Available    4/4/2006 9:05
PM    Not Available    Not Available    Not Available
sleep.exe      c:\webkit\sleep.exe 3020    8    204800 1413120    4/4/2006 9:06
PM    4.00    27.77 KB (28,432 bytes) 11/25/2005 1:10 PM

```

[Loaded Modules]

Name	Version	Size	File Date	Manufacturer	Path
winlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	497.00 KB (508,928 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\winlogon.exe
ntdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	748.50 KB (766,464 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\ntdll.dll
kernel32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,014.00 KB (1,038,336 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\kernel32.dll
advapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	605.50 KB (620,032 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\advapi32.dll
rpcrt4	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	627.00 KB (642,048 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\rpcrt4.dll
crypt32	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)	582.00 KB (595,968 bytes)			

	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\crypt32.dll				
msasn1	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		56.50 KB (57,856 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msasn1.dll	
msvert	7.0.3790.1830 (srv03_sp1_rtm.050324-1447)		340.50 KB (348,672 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msvert.dll	
user32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		574.50 KB (588,288 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\user32.dll	
gdi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		273.00 KB (279,552 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\gdi32.dll	
nddeapi	5.2.3790.0 (srv03_rtm.030324-2048)		16.00 KB (16,384 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\nddeapi.dll	
profmap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		22.50 KB (23,040 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\profmap.dll	
netapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		341.50 KB (349,696 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\netapi32.dll	
userenv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		771.00 KB (789,504 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\userenv.dll	
psapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		20.00 KB (20,480 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\psapi.dll	
regapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		55.00 KB (56,320 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\regapi.dll	
secur32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		64.00 KB (65,536 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\secur32.dll	
setupapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.03 MB (1,079,808 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\setupapi.dll	
version	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		18.00 KB (18,432 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\version.dll	
winsta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		54.50 KB (55,808 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\winsta.dll	
ws2_32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		82.00 KB (83,968 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\ws2_32.dll	
ws2help	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		19.50 KB (19,968 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\ws2help.dll	
msgina	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.16 MB (1,211,904 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msgina.dll	
shsvcs	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		131.50 KB (134,656 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\shsvcs.dll	
shlwapi	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		313.50 KB (321,024 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\shlwapi.dll	
sfc	5.2.3790.0 (srv03_rtm.030324-2048)		4.50 KB (4,608 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\sfc.dll	
sfc_os	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		138.00 KB (141,312 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\sfc_os.dll	
wintrust	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)		162.00 KB (165,888 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\wintrust.dll	
imagehlp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		145.50 KB (148,992 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\imagehlp.dll	
ole32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		1.19 MB (1,245,184 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\ole32.dll	
comctl32	6.0 (srv03_sp1_rtm.050324-1447)		1.00 MB (1,051,136 bytes)	3/24/2005 9:41 PM	Microsoft Corporation	d:\windows\winsxs\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.3790.1830_x-ww_7ae38ccf\comctl32.dll	

sxs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	743.50	KB	(761,344	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\sxs.dll			
winscard	5.2.3790.0 (srv03_rtm.030324-2048)	98.50	KB	(100,864	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\winscard.dll			
wtsapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	19.00	KB	(19,456	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\wtsapi32.dll			
shell32	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	7.99	MB	(8,379,392	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\shell32.dll			
rsaenh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	183.98	KB	(188,392	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\rsaenh.dll			
wldap32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	174.50	KB	(178,688	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\wldap32.dll			
cscdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	100.00	KB	(102,400	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\cscdll.dll			
dimsntfy	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	19.00	KB	(19,456	bytes)
	11/22/2005 4:22 PM Microsoft Corporation	d:\windows\system32\dimsntfy.dll			
wlnotify	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	94.50	KB	(96,768	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\wlnotify.dll			
mpr	5.2.3790.0 (srv03_rtm.030324-2048)	56.00	KB	(57,344	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\mpr.dll			
oleaut32	5.2.3790.1830	543.00	KB	(556,032	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\oleaut32.dll			
winmm	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	172.50	KB	(176,640	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\winmm.dll			
winspool	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	147.00	KB	(150,528	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\winspool.drv			
comctl32	5.82 (srv03_sp1_rtm.050324-1447)	585.00	KB	(599,040	bytes)
	3/24/2005 9:41 PM Microsoft Corporation	d:\windows\winsxs\x86_microsoft.windows.common-			
controls_6595b64144ccf1df_5.82.3790.1830_x-ww_1b6f474a\comctl32.dll					
uxtheme	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	202.00	KB	(206,848	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\uxtheme.dll			
samlib	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	46.50	KB	(47,616	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\samlib.dll			
clbcatq	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)	502.50	KB	(514,560	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\clbcatq.dll			
comres	2001.12.4720.0 (srv03_rtm.030324-2048)	778.00	KB	(796,672	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\comres.dll			
xpsp2res	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	2.76	MB	(2,897,920	bytes)
	11/22/2005 4:22 PM Microsoft Corporation	d:\windows\system32\xpsp2res.dll			
cscui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	319.50	KB	(327,168	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\cscui.dll			
drprov	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00	KB	(14,336	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\drprov.dll			
ntlanman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	43.50	KB	(44,544	bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\ntlanman.dll			
netui0	5.2.3790.0 (srv03_rtm.030324-2048)	75.50	KB	(77,312	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\netui0.dll			
netui1	5.2.3790.0 (srv03_rtm.030324-2048)	184.00	KB	(188,416	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\netui1.dll			
davclnt	5.2.3790.0 (srv03_rtm.030324-2048)	23.50	KB	(24,064	bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\davclnt.dll			
mprui	5.2.3790.0 (srv03_rtm.030324-2048)	49.00	KB	(50,176	bytes)
	3/25/2003 8:00 PM				

	Microsoft Corporation	d:\windows\system32\mprui.dll				
netui2	5.2.3790.0 (srv03_rtm.030324-2048)	309.50 KB (316,928 bytes)	3/25/2003	8:00		
PM	Microsoft Corporation	d:\windows\system32\netui2.dll				
comdlg32	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	274.50 KB (281,088 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\comdlg32.dll				
netmsg	5.2.3790.0 (srv03_rtm.030324-2048)	178.00 KB (182,272 bytes)	3/25/2003	8:00		
PM	Microsoft Corporation	d:\windows\system32\netmsg.dll				
ntmarta	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	120.50 KB (123,392 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\ntmarta.dll				
wbemprox	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	20.50 KB (20,992 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\wbem\wbemprox.dll				
wbemcomn	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	221.00 KB (226,304 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\wbem\wbemcomn.dll				
wbemsvc	5.2.3790.0 (srv03_rtm.030324-2048)	42.50 KB (43,520 bytes)	6/3/2005	2:50	PM	
	Microsoft Corporation	d:\windows\system32\wbem\wbemsvc.dll				
fastprox	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	471.00 KB (482,304 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\wbem\fastprox.dll				
msvcpx60	6.05.2144.0	388.00 KB (397,312 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\msvcpx60.dll				
ntdsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	71.00 KB (72,704 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\ntdsapi.dll				
dnsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	153.50 KB (157,184 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\dnsapi.dll				
services	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	107.50 KB (110,080 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\services.exe				
ncobjapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	36.00 KB (36,864 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\ncobjapi.dll				
scesrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	327.00 KB (334,848 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\scesrv.dll				
authz	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	66.50 KB (68,096 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\authz.dll				
umpnpgmr	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	126.50 KB (129,536 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\umpnpgmr.dll				
eventlog	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	67.50 KB (69,120 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\eventlog.dll				
lsass	5.2.3790.0 (srv03_rtm.030324-2048)	13.00 KB (13,312 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\lsass.exe				
lsasrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	803.00 KB (822,272 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\lsasrv.dll				
samsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	450.50 KB (461,312 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\samsrv.dll				
cryptdll	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	32.00 KB (32,768 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\cryptdll.dll				
msprivs	5.2.3790.0 (srv03_rtm.030324-2048)	46.50 KB (47,616 bytes)	3/25/2003	8:00	PM	
	Microsoft Corporation	d:\windows\system32\msprivs.dll				
kerberos	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	340.50 KB (348,672 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\kerberos.dll				
msv1_0	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	141.00 KB (144,384 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\msv1_0.dll				
iphlpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	92.50 KB (94,720 bytes)	11/22/2005	4:21	PM	
	Microsoft Corporation	d:\windows\system32\iphlpapi.dll				

netlogon	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	409.50 KB	(419,328 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\netlogon.dll	
w32time	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	222.00 KB	(227,328 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\w32time.dll	
schannel	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	141.00 KB	(144,384 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\schannel.dll	
wdigest	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	74.00 KB	(75,776 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\wdigest.dll	
rassfm	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	23.00 KB	(23,552 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\rassfm.dll	
kdcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	213.50 KB	(218,624 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\kdcsvc.dll	
ntdsa	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1.45 MB	(1,516,032 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\ntdsa.dll	
esent	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	1,022.50 KB	(1,047,040 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\esent.dll	
ntdsatq	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	29.50 KB	(30,208 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\ntdsatq.dll	
msswsock	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	250.50 KB	(256,512 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\msswsock.dll	
scecli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	186.50 KB	(190,976 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\scecli.dll	
ws03res	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	793.50 KB	(812,544 bytes)
	11/22/2005 4:22 PM Microsoft Corporation	d:\windows\system32\ws03res.dll	
ipsevcs	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	180.50 KB	(184,832 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\ipsevcs.dll	
oakley	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	264.00 KB	(270,336 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\oakley.dll	
winipsec	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	35.50 KB	(36,352 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\winipsec.dll	
pstorsvc	5.2.3790.0 (srv03_rtm.030324-2048)	24.00 KB	(24,576 bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\pstorsvc.dll	
psbase	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	84.00 KB	(86,016 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\psbase.dll	
hnetcfg	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	343.50 KB	(351,744 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\hnetcfg.dll	
wshtcpip	5.2.3790.0 (srv03_rtm.030324-2048)	18.00 KB	(18,432 bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\wshtcpip.dll	
dssenh	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	139.98 KB	(143,336 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\dssenh.dll	
wlbsctrl	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	82.00 KB	(83,968 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\wlbsctrl.dll	
w3ssl	6.0.3790.0 (srv03_rtm.030324-2048)	15.00 KB	(15,360 bytes)
	3/25/2003 8:00 PM Microsoft Corporation	d:\windows\system32\w3ssl.dll	
strmfilt	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	84.00 KB	(86,016 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\strmfilt.dll	
httpapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	24.00 KB	(24,576 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\httpapi.dll	
svchost	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	14.00 KB	(14,336 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\svchost.exe	
rpcss	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	406.00 KB	(415,744 bytes)
	11/22/2005 4:21 PM Microsoft Corporation	d:\windows\system32\rpcss.dll	
wzcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	364.50 KB	(373,248 bytes)

	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\wzcsvc.dll			
rtutils	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		34.50 KB (35,328 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\rtutils.dll			
wmi	5.2.3790.0 (srv03_rtm.030324-2048)		6.50 KB (6,656 bytes)	3/25/2003 8:00 PM		
		Microsoft Corporation	d:\windows\system32\wmi.dll			
dhcpcsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		113.50 KB (116,224 bytes)	3/25/2003 8:00 PM		
		Microsoft Corporation	d:\windows\system32\dhcpcsvc.dll			
atl	3.05.2283		83.00 KB (84,992 bytes)	3/25/2003 8:00 PM		Microsoft Corporation
			d:\windows\system32\atl.dll			
rastls	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		180.00 KB (184,320 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\rastls.dll			
cryptui	5.131.3790.1830 (srv03_sp1_rtm.050324-1447)		496.50 KB (508,416 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\cryptui.dll			
mprapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		89.00 KB (91,136 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\mprapi.dll			
activeds	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		194.00 KB (198,656 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\activeds.dll			
adslrpc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		146.00 KB (149,504 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\adslrpc.dll			
credui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		162.00 KB (165,888 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\credui.dll			
rasapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		239.50 KB (245,248 bytes)	3/25/2003 8:00 PM		
		Microsoft Corporation	d:\windows\system32\rasapi32.dll			
rasman	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		61.50 KB (62,976 bytes)	3/25/2003 8:00 PM		
		Microsoft Corporation	d:\windows\system32\rasman.dll			
tapi32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		179.50 KB (183,808 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\tapi32.dll			
raschap	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		119.50 KB (122,368 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\raschap.dll			
schedsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		197.50 KB (202,240 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\schedsvc.dll			
msidle	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)		6.50 KB (6,656 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\msidle.dll			
audiosrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		40.50 KB (41,472 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\audiosrv.dll			
wkssvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		130.00 KB (133,120 bytes)	3/25/2003 8:00 PM		
		Microsoft Corporation	d:\windows\system32\wkssvc.dll			
wiarpc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		32.50 KB (33,280 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\wiarpc.dll			
aelupsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		26.00 KB (26,624 bytes)	11/22/2005 4:22 PM		
		Microsoft Corporation	d:\windows\system32\aelupsvc.dll			
apphelp	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		146.50 KB (150,016 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\apphelp.dll			
cryptsvc	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		55.50 KB (56,832 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\cryptsvc.dll			
certcli	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		227.00 KB (232,448 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\certcli.dll			
vssapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		548.00 KB (561,152 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\vssapi.dll			
dmserver	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)		25.50 KB (26,112 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\dmserver.dll			
es	2001.12.4720.1830 (srv03_sp1_rtm.050324-1447)		233.00 KB (238,592 bytes)	11/22/2005 4:21 PM		
		Microsoft Corporation	d:\windows\system32\es.dll			

pchsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 39.00 KB (39,936 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\pchealth\helpctr\binaries\pchsvc.dll
srvsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\srvsvc.dll
seclogon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 18.50 KB (18,944 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\seclogon.dll
sens 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 36.50 KB (37,376 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\sens.dll
trkwks 5.2.3790.0 (srv03_rtm.030324-2048) 85.00 KB (87,040 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\trkwks.dll
wmisvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 140.00 KB (143,360 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\wmisvc.dll
wuauserv 5.7.3790.1830 (srv03_sp1_rtm.050324-1447) 8.00 KB (8,192 bytes) 11/22/2005 4:22 PM
Microsoft Corporation d:\windows\system32\wuauserv.dll
wuauieng 5.7.3790.1830 (srv03_sp1_rtm.050324-1447) 1.18 MB (1,232,896 bytes) 11/22/2005 4:22 PM
Microsoft Corporation d:\windows\system32\wuauieng.dll
advpack 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 98.00 KB (100,352 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\advpack.dll
cabinet 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 81.50 KB (83,456 bytes) 3/24/2005 8:35 PM
Microsoft Corporation d:\windows\system32\cabinet.dll
mspatcha 5.2.3790.0 (srv03_rtm.030324-2048) 29.00 KB (29,696 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\mspachtd.dll
shfolder 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 24.50 KB (25,088 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\shfolder.dll
winhttp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 353.00 KB (361,472 bytes) 3/24/2005 9:41 PM
Microsoft Corporation d:\windows\winsxs\x86_microsoft.windows.winhttp_6595b64144ccf1df_5.1.3790.1830_x-ww_74150efb\winhttp.dll
browser 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 76.50 KB (78,336 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\browser.dll
comsvcs 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 1.19 MB (1,248,256 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\comsvcs.dll
wbemcore 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 497.50 KB (509,440 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\wbemcore.dll
esscli 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 250.00 KB (256,000 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\esscli.dll
wmiutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 93.50 KB (95,744 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\wmiutils.dll
repdrvfs 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 172.50 KB (176,640 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\repdrvfs.dll
wmiprvsd 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 404.00 KB (413,696 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\wmiprvsd.dll
wbemess 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 271.50 KB (278,016 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\wbem\wbemess.dll
netrap 5.2.3790.0 (srv03_rtm.030324-2048) 11.50 KB (11,776 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\netrap.dll
netman 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 258.50 KB (264,704 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\netman.dll
netshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.73 MB (1,812,992 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\netshell.dll
clusapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 60.00 KB (61,440 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\clusapi.dll

wininet 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 646.00 KB (661,504 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\wininet.dll

wzcsapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 41.00 KB (41,984 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\wzcsapi.dll

rasdlg 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 663.00 KB (678,912 bytes)
3/25/2003 8:00 PM Microsoft Corporation d:\windows\system32\rasdlg.dll

rasadhlp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 7.50 KB (7,680 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\rasadhlp.dll

ncprov 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 46.50 KB (47,616 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\wbem\ncprov.dll

ntlsapi 5.2.3790.0 (srv03_rtm.030324-2048) 8.00 KB (8,192 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\ntlsapi.dll

spoolsv 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 57.00 KB (58,368 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\spoolsv.exe

spoolss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 85.00 KB (87,040 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\spoolss.dll

localspl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 339.00 KB (347,136 bytes)
3/25/2003 8:00 PM Microsoft Corporation d:\windows\system32\localspl.dll

cnbjmon 5.2.3790.1224 (dnsvr(skatar).040514-1058) 46.50 KB (47,616 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\cnbjmon.dll

pjlmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 15.00 KB (15,360 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\pjlmon.dll

tcpmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 47.00 KB (48,128 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\tcpmon.dll

wsnmp32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 43.00 KB (44,032 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\wsnmp32.dll

tcpmib 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.50 KB (17,920 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\tcpmib.dll

wsock32 5.2.3790.0 (srv03_rtm.030324-2048) 22.00 KB (22,528 bytes) 3/25/2003 8:00
PM Microsoft Corporation d:\windows\system32\wsock32.dll

mgmtapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 15.50 KB (15,872 bytes)
3/25/2003 8:00 PM Microsoft Corporation d:\windows\system32\mgmtapi.dll

snmpapi 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 19.50 KB (19,968 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\snmpapi.dll

usbmon 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.00 KB (17,408 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\usbmon.dll

winrnr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 17.00 KB (17,408 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\winrnr.dll

wshqos 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\wshqos.dll

win32spl 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 100.50 KB (102,912 bytes)
3/25/2003 8:00 PM Microsoft Corporation d:\windows\system32\win32spl.dll

inetpp 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 75.00 KB (76,800 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetpp.dll

icmp 5.2.3790.0 (srv03_rtm.030324-2048) 4.50 KB (4,608 bytes) 3/25/2003 8:00 PM
Microsoft Corporation d:\windows\system32\icmp.dll

ersvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 24.00 KB (24,576 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\ersvc.dll

inetinfo 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 14.00 KB (14,336 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetsrv\inetinfo.exe

iisutil 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 164.00 KB (167,936 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\inetsrv\iisutil.dll

rpcref 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 4.00 KB (4,096 bytes) 11/22/2005 4:21

PM	Microsoft Corporation	d:\windows\system32\inetsrv\rpcpref.dll				
iisrtl	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	138.50 KB	(141,824 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\iisrtl.dll			
iisadmin	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	21.00 KB	(21,504 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\iisadmin.dll			
coadmin	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	62.50 KB	(64,000 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\coadmin.dll			
admwprox	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	47.00 KB	(48,128 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\admwprox.dll			
iiscfg	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	1.08 MB	(1,133,056 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\iiscfg.dll			
metadata	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	229.00 KB	(234,496 bytes)			
	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\metadata.dll			
msxml3	8.70.1104.0	1.06 MB	(1,107,456 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msxml3.dll
svcxext	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	43.50 KB	(44,544 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\svcxext.dll
security	5.2.3790.0 (srv03_rtm.030324-2048)	5.50 KB	(5,632 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\security.dll
iismap	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	58.50 KB	(59,904 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\iismap.dll
wamreg	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	54.50 KB	(55,808 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\wamreg.dll
iisw3adm	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	211.00 KB	(216,064 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\iisw3adm.dll
w3cache	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	19.00 KB	(19,456 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\w3cache.dll
w3tp	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	13.00 KB	(13,312 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\w3tp.dll
lonsint	6.0.3790.1830 (srv03_sp1_rtm.050324-1447)	13.00 KB	(13,312 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\inetsrv\lonsint.dll
termsrv	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	239.00 KB	(244,736 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\termsrv.dll
icaapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	12.50 KB	(12,800 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\icaapi.dll
mstlsapi	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	116.00 KB	(118,784 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\mstlsapi.dll
rdpwsx	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	101.63 KB	(104,072 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\rdpwsx.dll
explorer	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.00 MB	(1,050,624 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\explorer.exe
browseui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1,009.00 KB	(1,033,216 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\browseui.dll
shdocvw	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	1.43 MB	(1,502,720 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\shdocvw.dll
themeui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	377.50 KB	(386,560 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\themeui.dll
msimg32	5.2.3790.0 (srv03_rtm.030324-2048)	4.50 KB	(4,608 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\msimg32.dll
actxprxy	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	96.50 KB	(98,816 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\actxprxy.dll
linkinfo	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	19.00 KB	(19,456 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\linkinfo.dll

PM	Microsoft Corporation	d:\windows\system32\linkinfo.dll						
ntshrui	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	140.00 KB (143,360 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\ntshrui.dll			
webcheck	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	272.50 KB (279,040 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\webcheck.dll			
stobject	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	120.50 KB (123,392 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\stobject.dll			
batmeter	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	31.50 KB (32,256 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\batmeter.dll			
powrprof	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	16.50 KB (16,896 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\powrprof.dll			
cmd	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	379.00 KB (388,096 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\cmd.exe			
rdpsnd	5.2.3790.0 (srv03_rtm.030324-2048)	18.00 KB (18,432 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\rdpsnd.dll			
scredir	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	28.00 KB (28,672 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\scredir.dll			
msacm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	22.00 KB (22,528 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msacm32.drv			
msacm32	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	69.50 KB (71,168 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msacm32.dll			
imaadp32	5.2.3790.0 (srv03_rtm.030324-2048)	15.50 KB (15,872 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\imaadp32.acm			
msadp32	5.2.3790.0 (srv03_rtm.030324-2048)	14.50 KB (14,848 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\msadp32.acm			
msg711	5.2.3790.0 (srv03_rtm.030324-2048)	10.00 KB (10,240 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\msg711.acm			
msgsm32	5.2.3790.0 (srv03_rtm.030324-2048)	20.50 KB (20,992 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\msgsm32.acm			
printui	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	563.00 KB (576,512 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\printui.dll			
cfgmgr32	5.2.3790.0 (srv03_rtm.030324-2048)	17.50 KB (17,920 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\cfgmgr32.dll			
tssoft32	1.01	9.50 KB (9,728 bytes)	3/25/2003 8:00 PM		d:\windows\system32\tssoft32.acm	DSP	GROUP,	INC.
tsd32	1.03	16.50 KB (16,896 bytes)	3/25/2003 8:00 PM		d:\windows\system32\tsd32.dll	DSP	GROUP,	INC.
msg723	5.2.3790.1830	120.00 KB (122,880 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\msg723.acm			
msaud32	8.00.00.4487	288.00 KB (294,912 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\msaud32.acm			
sl_anet	3.02	84.00 KB (86,016 bytes)	3/25/2003 8:00 PM	Sipro Lab Telecom Inc.	d:\windows\system32\sl_anet.acm			
l3codeca	1, 9, 0, 0305	284.00 KB (290,816 bytes)	3/25/2003 8:00 PM	Fraunhofer Institut Integrierte Schaltungen IIS	d:\windows\system32\l3codeca.acm			
rdpclip	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	68.00 KB (69,632 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\rdpclip.exe			
urlmon	6.00.3790.1830 (srv03_sp1_rtm.050324-1447)	673.00 KB (689,152 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\urlmon.dll			
browselc	6.00.3790.0 (srv03_rtm.030324-2048)	62.00 KB (63,488 bytes)	3/25/2003 8:00 PM	Microsoft Corporation	d:\windows\system32\browselc.dll			
duser	5.2.3790.1830 (srv03_sp1_rtm.050324-1447)	295.00 KB (302,080 bytes)	11/22/2005 4:21 PM	Microsoft Corporation	d:\windows\system32\duser.dll			

```

mlang 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 577.50 KB (591,360 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\mlang.dll
shdoclc 6.00.3790.0 (srv03_rtm.030324-2048) 588.50 KB (602,624 bytes) 3/25/2003 8:00
PM Microsoft Corporation d:\windows\system32\shdoclc.dll
gdiplus 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 1.64 MB (1,724,416 bytes)
3/24/2005 9:41 PM Microsoft Corporation
d:\windows\winsxs\x86_microsoft.windows.gdiplus_6595b64144ccf1df_1.0.3790.1830_x-
ww_24c40c58\gdiplus.dll
w3wp 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 7.00 KB (7,168 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetsrv\w3wp.exe
w3core 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 340.50 KB (348,672 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\inetsrv\w3core.dll
w3comlog 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 10.50 KB (10,752 bytes)
11/22/2005 4:21 PM Microsoft Corporation
d:\windows\system32\inetsrv\w3comlog.dll
w3dt 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 38.50 KB (39,424 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetsrv\w3dt.dll
iisres 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 120.00 KB (122,880 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\inetsrv\iisres.dll
aspnet_filter 2.0.50727.42 (RTM.050727-4200) 10.50 KB (10,752 bytes) 9/23/2005 7:28
AM Microsoft Corporation d:\windows\microsoft.net\framework\v2.0.50727\aspnet_filter.dll
msvcr80 8.00.50727.42 612.00 KB (626,688 bytes) 9/23/2005 7:29 AM
Microsoft Corporation
d:\windows\winsxs\x86_microsoft.vc80.crt_1fc8b3b9a1e18e3b_8.0.50727.42_x-
ww_0de06acd\msvcr80.dll
w3isapi 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 61.00 KB (62,464 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetsrv\w3isapi.dll
gzip 6.0.3790.1830 (srv03_sp1_rtm.050324-1447) 25.00 KB (25,600 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\inetsrv\gzip.dll
""\d:\inetpub\wwwroot\tpcc.dll" ""\d:\inetpub\wwwroot\tpcc.dll"
msvcr70 7.00.9466.0 336.00 KB (344,064 bytes) 6/3/2005 3:26 PM
Microsoft Corporation d:\windows\system32\msvcr70.dll
tpcc_com Not Available 10.50 KB (10,752 bytes) 6/3/2005 3:26 PM Not Available
d:\inetpub\wwwroot\tpcc_com.dll
tpcc_odbc Not Available 20.00 KB (20,480 bytes) 6/3/2005 3:26 PM Not Available
d:\inetpub\wwwroot\tpcc_odbc.dll
odbc32 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 240.00 KB (245,760 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\odbc32.dll
odbcint 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 92.00 KB (94,208 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\odbcint.dll
sqlsrv322000.086.1830.00 (srv03_sp1_rtm.050324-1447) 436.00 KB (446,464 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\sqlsrv32.dll
sqlunirl 2000.080.0728.00 176.56 KB (180,800 bytes) 11/22/2005 4:21 PM
Microsoft Corporation d:\windows\system32\sqlunirl.dll
sqlsrv322000.086.1830.00 (srv03_sp1_rtm.050324-1447) 88.00 KB (90,112 bytes) 11/22/2005 4:21
PM Microsoft Corporation d:\windows\system32\sqlsrv32.rll
odbccp32 3.526.1830.0 (srv03_sp1_rtm.050324-1447) 100.00 KB (102,400 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\odbccp32.dll
dbnetlib 2000.086.1830 (srv03_sp1_rtm.050324-1447) 112.00 KB (114,688 bytes)
11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\dbnetlib.dll
tpcc_com_all 1, 0, 0, 1 104.00 KB (106,496 bytes) 6/3/2005 3:26 PM
d:\inetpub\wwwroot\tpcc_com_all.dll
dllhost 5.2.3790.0 (srv03_rtm.030324-2048) 5.50 KB (5,632 bytes) 3/25/2003 8:00 PM

```

Microsoft Corporation d:\windows\system32\dlhhost.exe

txflog 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 96.50 KB (98,816 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\txflog.dll

xolehlp 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 10.50 KB (10,752 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\xolehlp.dll

msdtcprx 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 455.50 KB (466,432 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\msdtcprx.dll

mtxclu 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 77.00 KB (78,848 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\mtxclu.dll

resutils 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 63.50 KB (65,024 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\resutils.dll

catsrv 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 273.00 KB (279,552 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\catsrv.dll

clbcatex 2001.12.4720.1830 (srv03_sp1_rtm.050324-1447) 102.50 KB (104,960 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\clbcatex.dll

helpctr 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 778.00 KB (796,672 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\pchealth\helpctr\binaries\helpctr.exe

hcappres 5.2.3790.0 (srv03_rtm.030324-2048) 6.50 KB (6,656 bytes) 6/3/2005 2:53 PM Microsoft Corporation d:\windows\pchealth\helpctr\binaries\hcappres.dll

itss 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 133.50 KB (136,704 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\itss.dll

pchshell 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 104.50 KB (107,008 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\pchealth\helpctr\binaries\pchshell.dll

mshtml 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 2.96 MB (3,108,864 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\mshtml.dll

msls31 3.10.349.0 142.00 KB (145,408 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\msls31.dll

msimtf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 156.00 KB (159,744 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\msimtf.dll

msctf 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 311.00 KB (318,464 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\msctf.dll

jscrip 5.6.0.8827 448.00 KB (458,752 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\jscrip.dll

imm32 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 108.00 KB (110,592 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\imm32.dll

mshtml 6.00.3790.1830 (srv03_sp1_rtm.050324-1447) 454.50 KB (465,408 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\mshtml.dll

vbscript 5.6.0.8827 392.00 KB (401,408 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\vbscript.dll

msinfo 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 376.00 KB (385,024 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\pchealth\helpctr\binaries\msinfo.dll

mfc42u 6.06.8063.0 1.11 MB (1,163,776 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\mfc42u.dll

riched32 5.2.3790.0 (srv03_rtm.030324-2048) 3.50 KB (3,584 bytes) 3/25/2003 8:00 PM Microsoft Corporation d:\windows\system32\riched32.dll

riched20 5.31.23.1224 439.00 KB (449,536 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\system32\riched20.dll

helpsvc 5.2.3790.1830 (srv03_sp1_rtm.050324-1447) 745.00 KB (762,880 bytes) 11/22/2005 4:21 PM Microsoft Corporation d:\windows\pchealth\helpctr\binaries\helpsvc.exe

sleep 4.00 27.77 KB (28,432 bytes) 11/25/2005 1:10 PM Microsoft Corporation
 c:\webkit\sleep.exe

[Services]

Display Name	Name	State	Start Mode	Service Type	Path	Error Control	Start
Name	Tag ID						
Application Experience Lookup Service	AeLookupSvc	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Alerters	Alerters	Stopped	Disabled	Share	Process		
localservice	Normal NT AUTHORITY\LocalService		0				-k
Application Layer Gateway Service	ALG	Stopped	Manual	Own	Process		
	d:\windows\system32\alg.exe		Normal	NT AUTHORITY\LocalService	0		
Application Management AppMgmt		Stopped	Manual	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
ASP.NET State Service	aspnet_state	Stopped	Manual	Own	Process		
	d:\windows\microsoft.net\framework\v2.0.50727\aspnet_state.exe		Normal	NT AUTHORITY\NetworkService	0		
Windows Audio	AudioSrv	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Background Intelligent Transfer Service	BITS	Stopped	Manual	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Computer Browser	Browser	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
Indexing Service	CiSvc	Stopped	Disabled	Share	Process		
	d:\windows\system32\cisvc.exe		Normal	LocalSystem	0		
ClipBook	ClipSrv	Stopped	Disabled	Own	Process		
	d:\windows\system32\clipsrv.exe		Normal	LocalSystem	0		
.NET Runtime Optimization Service	v2.0.50727_X86	Stopped	Manual	Own	Process		
	d:\windows\microsoft.net\framework\v2.0.50727\mscorsvw.exe		Ignore	LocalSystem	0		
COM+ System Application	COMSysApp	Running	Manual	Own	Process		
	d:\windows\system32\dllhost.exe /processid:{02d4b3f1-fd88-11d1-960d-00805fc79235}		Normal	LocalSystem	0		
Cryptographic Services	CryptSvc	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
DCOM Server Process Launcher	DcomLaunch	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k dcomlaunch		Normal	LocalSystem	0		
Distributed File System	Dfs	Stopped	Manual	Own	Process		
	d:\windows\system32\dfssvc.exe		Normal	LocalSystem	0		
DHCP Client	Dhcp	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k networkservice		Normal	NT AUTHORITY\NetworkService	0		
Logical Disk Manager Administrative Service	dmadmin	Stopped	Manual	Share	Process		
	d:\windows\system32\dmadmin.exe /com		Normal	LocalSystem	0		
Logical Disk Manager	dmserver	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0		
DNS Client	Dnscache	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k networkservice		Normal	NT AUTHORITY\NetworkService	0		
Error Reporting Service	ERSvc	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe -k winerr		Ignore	LocalSystem	0		
Event Log	Eventlog	Running	Auto	Share	Process		
	d:\windows\system32\services.exe		Normal	LocalSystem	0		

COM+ Event System	EventSystem	Running	Auto	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
Help and Support	helpsvc	Running	Auto	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
Human Interface Device Access	HidServ	Stopped	Disabled	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
HTTP SSL	HTTPFilter	Running	Manual	Share		Process
	d:\windows\system32\lsass.exe		Normal	LocalSystem	0	
IIS Admin Service	IISADMIN	Running	Auto	Share		Process
	d:\windows\system32\inetrv\inetinfo.exe		Normal	LocalSystem	0	
IMAPI CD-Burning COM Service	ImapiService	Stopped	Disabled		Own	Process
	d:\windows\system32\imapi.exe		Normal	LocalSystem	0	
Intersite Messaging	IsmServ	Stopped	Disabled		Own	Process
	d:\windows\system32\ismserv.exe		Normal	LocalSystem	0	
Kerberos Key Distribution Center	kdc	Stopped	Disabled	Share		Process
	d:\windows\system32\lsass.exe		Normal	LocalSystem	0	
Server lanmanserver		Running	Auto	Share	Process	d:\windows\system32\svchost.exe
-k netsvcs			Normal	LocalSystem	0	
Workstation	lanmanworkstation	Running	Auto	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
License Logging	LicenseService	Stopped	Disabled		Own	Process
	d:\windows\system32\llssrv.exe		Normal	NT AUTHORITY\NetworkService	0	
TCP/IP NetBIOS Helper	LmHosts	Running	Auto	Share		Process
	d:\windows\system32\svchost.exe -k localservice		Normal	NT AUTHORITY\LocalService	0	
Messenger	Messenger	Stopped	Disabled	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
NetMeeting Remote Desktop Sharing	mnmsrvc	Stopped	Disabled		Own	Process
	d:\windows\system32\mnmsrvc.exe		Normal	LocalSystem	0	
Distributed Transaction Coordinator	MSDTC	Running	Auto		Own	Process
	d:\windows\system32\msdtc.exe		Normal	NT AUTHORITY\NetworkService	0	
Windows Installer	MSIServer	Stopped	Manual	Share		Process
	d:\windows\system32\msiexec.exe /v		Normal	LocalSystem	0	
Visual Studio 2005 Remote Debugger	msvsmon80	Stopped	Disabled		Own	Process
	"d:\program files\microsoft visual studio 8\common7\ide\remote debugger\x86\msvsmon.exe" /service msvsmon80		Ignore	LocalSystem	0	
Network DDE	NetDDE	Stopped	Disabled	Share		Process
	d:\windows\system32\netdde.exe		Normal	LocalSystem	0	
Network DDE DSDM	NetDDEdsdm	Stopped	Disabled	Share		Process
	d:\windows\system32\netdde.exe		Normal	LocalSystem	0	
Net Logon	Netlogon	Stopped	Manual	Share	Process	d:\windows\system32\lsass.exe
			Normal	LocalSystem	0	
Network Connections	Netman	Running	Manual	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
Network Location Awareness (NLA)	Nla	Running	Manual	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
File Replication	NtFrs	Stopped	Manual	Own	Process	d:\windows\system32\ntfrs.exe
			LocalSystem	0		
NT LM Security Support Provider	NtLmSsp	Stopped	Manual	Share		Process
	d:\windows\system32\lsass.exe		Normal	LocalSystem	0	
Removable Storage	NtmsSvc	Stopped	Manual	Share		Process
	d:\windows\system32\svchost.exe -k netsvcs		Normal	LocalSystem	0	
Office Source Engine	ose	Stopped	Manual	Own	Process	"d:\program files\common

files\microsoft shared\source engine\ose.exe"	Normal	LocalSystem	0			
Plug and Play PlugPlay Running	Auto	Share				Process
d:\windows\system32\services.exe	Normal	LocalSystem	0			
IPSEC Services PolicyAgent Running	Auto	Share				Process
d:\windows\system32\lsass.exe	Normal	LocalSystem	0			
Protected Storage ProtectedStorageRunning	Auto	Share				Process
d:\windows\system32\lsass.exe	Normal	LocalSystem	0			
Remote Access Auto Connection Manager RasAuto	Stopped	Manual	Share			Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Remote Access Connection Manager RasMan	Stopped	Manual	Share			Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Remote Desktop Help Session Manager RDSessMgr	Stopped	Manual	Own			Process
d:\windows\system32\sessmgr.exe	Normal	LocalSystem	0			
Routing and Remote Access RemoteAccess	Stopped	Disabled	Share			Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Remote Registry RemoteRegistry Running	Auto	Share				Process
d:\windows\system32\svchost.exe -k regsvc	Normal	NT AUTHORITY\LocalService	0			
Remote Command Service RMSYSRunning	Auto	Own	Process	"d:\program		
files\benchcraft\rsys.exe" Normal	\Administrator	0				
Remote Procedure Call (RPC) Locator RpcLocator	Stopped	Manual	Own			Process
d:\windows\system32\locator.exe	Normal	NT AUTHORITY\NetworkService	0			
Remote Procedure Call (RPC) RpcSs Running	Auto	Share				Process
d:\windows\system32\svchost.exe -k rpcss	Normal	NT Authority\NetworkService	0			
Resultant Set of Policy Provider RSoPProv	Stopped	Manual	Share			Process
d:\windows\system32\rsopprov.exe	Normal	LocalSystem	0			
Special Administration Console Helper sacsvr	Stopped	Manual	Share			Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Security Accounts Manager SamSs Running	Auto	Share				Process
d:\windows\system32\lsass.exe	Normal	LocalSystem	0			
Smart Card SCardSvr	Stopped	Manual	Share			Process
d:\windows\system32\scardsvr.exe	Ignore	NT AUTHORITY\LocalService	0			
Task Scheduler Schedule Running	Auto	Share				Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Secondary Logon seclogon Running	Auto	Share				Process
d:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem	0			
System Event Notification SENS Running	Auto	Share				Process
d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Windows Firewall/Internet Connection Sharing (ICS)	SharedAccess	Stopped	Disabled			
Share Process d:\windows\system32\svchost.exe -k netsvcs	Normal	LocalSystem	0			
Shell Hardware Detection ShellHWDetection	Running	Auto	Share			Process
d:\windows\system32\svchost.exe -k netsvcs	Ignore	LocalSystem	0			
Print Spooler Spooler Running	Auto	Own	Process	d:\windows\system32\spoolsv.exe		
Normal LocalSystem	0					
Windows Image Acquisition (WIA) stisvc	Stopped	Disabled	Share			Process
d:\windows\system32\svchost.exe -k imgsvc	Normal	NT AUTHORITY\LocalService	0			
Microsoft Software Shadow Copy Provider swprv	Stopped	Manual	Own			Process
d:\windows\system32\svchost.exe -k swprv	Normal	LocalSystem	0			
Performance Logs and Alerts SysmonLog	Stopped	Manual	Own			Process
d:\windows\system32\smlogsvc.exe	Normal	NT Authority\NetworkService	0			
Telephony TapiSrv	Stopped	Manual	Share	Process	d:\windows\system32\svchost.exe -k	

tapisrv	Normal	LocalSystem	0				
Terminal Services	TermService	Running	Manual	Share			Process
	d:\windows\system32\svchost.exe	-k termsvcs	Normal	LocalSystem	0		
Themes	Themes	Stopped	Disabled	Share	Process	d:\windows\system32\svchost.exe	-k
netsh	Normal	LocalSystem	0				
Telnet	TlntSvr	Stopped	Disabled	Own	Process	d:\windows\system32\tlntsvr.exe	Normal
	NT AUTHORITY\LocalService	0					
Distributed Link Tracking Server	TrkSvr	Stopped	Disabled	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
Distributed Link Tracking Client	TrkWks	Running	Auto	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
Terminal Services Session Directory	Tssdis	Stopped	Disabled	Own			Process
	d:\windows\system32\tssdis.exe	Normal	LocalSystem	0			
Windows User Mode Driver Framework	UMWdf	Stopped	Manual	Own			Process
	d:\windows\system32\wdfmgr.exe	Normal	NT AUTHORITY\LocalService	0			
Upload Manager	uploadmgr	Stopped	Manual	Share	Process	d:\windows\system32\svchost.exe	
	-k netsvcs	Normal	LocalSystem	0			
Uninterruptible Power Supply	UPS	Stopped	Manual	Own			Process
	d:\windows\system32\ups.exe	Normal	NT AUTHORITY\LocalService	0			
Virtual Disk Service	vds	Stopped	Manual	Own	Process	d:\windows\system32\vds.exe	
	Normal	LocalSystem	0				
Volume Shadow Copy	VSS	Stopped	Manual	Own	Process	d:\windows\system32\vssvc.exe	
	Normal	LocalSystem	0				
Windows Time	W32Time	Running	Auto	Share			Process
	d:\windows\system32\svchost.exe	-k localservice	Normal	NT AUTHORITY\LocalService	0		
World Wide Web Publishing Service	W3SVC	Running	Auto	Share			Process
	d:\windows\system32\svchost.exe	-k iissvcs	Normal	LocalSystem	0		
WebClient	WebClient	Stopped	Disabled	Share			Process
	d:\windows\system32\svchost.exe	-k localservice	Normal	NT AUTHORITY\LocalService	0		
WinHTTP Web Proxy Auto-Discovery Service	WinHttpAutoProxySvc	Stopped	Manual	Share			Process
	d:\windows\system32\svchost.exe	-k localservice	Normal	NT AUTHORITY\LocalService	0		
Windows Management Instrumentation	winmgmt	Running	Auto	Share	Process		
	d:\windows\system32\svchost.exe	-k netsvcs	Ignore	LocalSystem	0		
Portable Media Serial Number Service	WmdmPmSN	Stopped	Manual	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
Windows Management Instrumentation Driver Extensions	Wmi	Stopped	Manual	Share	Process		
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
WMI Performance Adapter	WmiApSrv	Stopped	Manual	Own			Process
	d:\windows\system32\wbem\wmiapsrv.exe	Normal	LocalSystem	0			
Automatic Updates	wuauclt	Running	Auto	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
Wireless Configuration	WZCSVC	Running	Auto	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		
Network Provisioning Service	xmlprov	Stopped	Manual	Share			Process
	d:\windows\system32\svchost.exe	-k netsvcs	Normal	LocalSystem	0		

[Program Groups]

Group Name	Name	User Name
Accessories	Default User:Accessories	Default User

Accessories\AccessibilityDefault User:Accessories\Accessibility Default User
 Accessories\Entertainment Default User:Accessories\Entertainment Default User
 Startup Default User:Startup Default User
 Accessories All Users:Accessories All Users
 Accessories\AccessibilityAll Users:Accessories\Accessibility All Users
 Accessories\Communications All Users:Accessories\Communications All Users
 Accessories\Entertainment All Users:Accessories\Entertainment All Users
 Accessories\System Tools All Users:Accessories\System Tools All Users
 Administrative Tools All Users:Administrative Tools All Users
 Microsoft SQL Server 2005 All Users:Microsoft SQL Server 2005 All Users
 Microsoft SQL Server 2005\Analysis Services All Users:Microsoft SQL Server 2005\Analysis ServicesAll Users
 Microsoft SQL Server 2005\Configuration Tools All Users:Microsoft SQL Server 2005\Configuration Tools All Users
 Microsoft SQL Server 2005\Documentation and Tutorials All Users:Microsoft SQL Server 2005\Documentation and Tutorials All Users
 Microsoft SQL Server 2005\Documentation and Tutorials\Tutorials All Users:Microsoft SQL Server 2005\Documentation and Tutorials\Tutorials All Users
 Microsoft SQL Server 2005\Performance Tools All Users:Microsoft SQL Server 2005\Performance Tools All Users
 Microsoft Visual Studio 2005 All Users:Microsoft Visual Studio 2005 All Users
 Microsoft Visual Studio 2005\Visual Studio Tools All Users:Microsoft Visual Studio 2005\Visual Studio Tools All Users
 Startup All Users:Startup All Users
 Accessories NT AUTHORITY\SYSTEM:Accessories NT AUTHORITY\SYSTEM
 Accessories\AccessibilityNT AUTHORITY\SYSTEM:Accessories\Accessibility NT AUTHORITY\SYSTEM
 Accessories\Entertainment NT AUTHORITY\SYSTEM:Accessories\Entertainment NT AUTHORITY\SYSTEM
 Startup NT AUTHORITY\SYSTEM:Startup NT AUTHORITY\SYSTEM
 Accessories WEB171\Administrator:Accessories WEB171\Administrator
 Accessories\AccessibilityWEB171\Administrator:Accessories\Accessibility WEB171\Administrator
 Accessories\Entertainment WEB171\Administrator:Accessories\Entertainment WEB171\Administrator
 Administrative Tools WEB171\Administrator:Administrative Tools WEB171\Administrator
 Benchcraft WEB171\Administrator:Benchcraft WEB171\Administrator
 Startup WEB171\Administrator:Startup WEB171\Administrator

[Startup Programs]

Program	Command	User Name	Location
desktop	desktop.ini	NT AUTHORITY\SYSTEM	Startup
desktop	desktop.ini	WEB171\Administrator	Startup
desktop	desktop.ini	.DEFAULT	Startup
desktop	desktop.ini	All Users	Common Startup

[OLE Registration]

Object	Local Server
Sound (OLE2)	sndrec32.exe
Media Clip	mplay32.exe
Video Clip	mplay32.exe /avi
MIDI Sequence	mplay32.exe /mid

Sound Not Available
 Media Clip Not Available
 WordPad Document "%programfiles%\windows nt\accessories\wordpad.exe"
 Windows Media Services DRM Storage object Not Available
 DDSContainerCtl Class Not Available
 Bitmap Image mspaint.exe

[Windows Error Reporting]

Time Type Details

[Internet Settings]

[Internet Explorer]

[Following are sub-categories of this main category]
 [Summary]

Item Value
 Version 6.0.3790.1830
 Build 63790.1830
 Application Path D:\Program Files\Internet Explorer
 Language English (United States)
 Active Printer Not Available

Cipher Strength 128-bit
 Content Advisor Disabled
 IEAK Install No

[File Versions]

File	Version	Size	Date	Path	Company
actxprxy.dll	6.0.3790.1830	97 KB	3/24/2005 5:55:26 PM	D:\WINDOWS\system32	Microsoft Corporation
advpack.dll	6.0.3790.1830	98 KB	3/24/2005 5:55:28 PM	D:\WINDOWS\system32	Microsoft Corporation
asctrls.ocx	6.0.3790.0	90 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	Microsoft Corporation
browsecl.dll	6.0.3790.0	62 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	Microsoft Corporation
browseui.dll	6.0.3790.1830	1,009 KB	3/24/2005 5:56:10 PM	D:\WINDOWS\system32	Microsoft Corporation
cdfview.dll	6.0.3790.1830	149 KB	3/24/2005 5:56:32 PM	D:\WINDOWS\system32	Microsoft Corporation
comctl32.dll	5.82.3790.1830	585 KB	3/24/2005 5:57:56 PM	D:\WINDOWS\system32	Microsoft Corporation
dxttrans.dll	6.3.3790.1830	205 KB	3/24/2005 6:00:58 PM	D:\WINDOWS\system32	Microsoft Corporation
dxtmsft.dll	6.3.3790.1830	355 KB	3/24/2005 6:00:58 PM	D:\WINDOWS\system32	Microsoft Corporation
iecont.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available

iecontlc.dll	<File Missing>	Not Available	Not Available	Not Available	Not Available
iedkcs32.dll	16.0.3790.1830 Microsoft Corporation	324 KB	3/24/2005 6:04:58 PM	D:\WINDOWS\system32	
iepeers.dll	6.0.3790.1830 Microsoft Corporation	248 KB	3/24/2005 6:04:58 PM	D:\WINDOWS\system32	
iesetup.dll	6.0.3790.1830 Microsoft Corporation	61 KB	3/24/2005 6:04:58 PM	D:\WINDOWS\system32	
ieunit.inf	Not Available	24 KB	3/24/2005 6:04:58 PM	D:\WINDOWS\system32	Not Available
ieexplore.exe	6.0.3790.1830 Microsoft Corporation	92 KB	3/24/2005 6:04:58 PM	D:\Program Files\Internet Explorer	
imgutil.dll	6.0.3790.1830 Microsoft Corporation	38 KB	3/24/2005 6:05:04 PM	D:\WINDOWS\system32	
inetctl.dll	6.0.3790.1830 Microsoft Corporation	358 KB	3/24/2005 6:05:06 PM	D:\WINDOWS\system32	
inetpl.dll	6.0.3790.0 Microsoft Corporation	109 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	
inseng.dll	6.0.3790.1830 Microsoft Corporation	94 KB	3/24/2005 6:05:06 PM	D:\WINDOWS\system32	
mlang.dll	6.0.3790.1830 Microsoft Corporation	578 KB	3/24/2005 6:07:20 PM	D:\WINDOWS\system32	
msencode.dll	2002.10.4.0 Microsoft Corporation	112 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	Microsoft Corporation
mshta.exe	6.0.3790.1830 Microsoft Corporation	30 KB	3/24/2005 6:07:26 PM	D:\WINDOWS\system32	
mshtml.dll	6.0.3790.1830 Microsoft Corporation	3,036 KB	3/24/2005 6:07:26 PM	D:\WINDOWS\system32	
mshtml.tlb	6.0.3790.1830 Microsoft Corporation	1,320 KB	3/24/2005 6:07:26 PM	D:\WINDOWS\system32	
mshtml.dll	6.0.3790.1830 Microsoft Corporation	455 KB	3/24/2005 6:07:26 PM	D:\WINDOWS\system32	
mshtml.dll	6.0.3790.1830 Microsoft Corporation	56 KB	3/24/2005 6:07:26 PM	D:\WINDOWS\system32	
msident.dll	6.0.3790.1830 Microsoft Corporation	48 KB	3/24/2005 6:07:28 PM	D:\WINDOWS\system32	
msidntld.dll	6.0.3790.0 Microsoft Corporation	15 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	
msieftp.dll	6.0.3790.1830 Microsoft Corporation	244 KB	3/24/2005 6:07:28 PM	D:\WINDOWS\system32	
msrating.dll	6.0.3790.1830 Microsoft Corporation	144 KB	3/24/2005 6:07:36 PM	D:\WINDOWS\system32	
mstime.dll	6.0.3790.1830 Microsoft Corporation	523 KB	3/24/2005 6:07:38 PM	D:\WINDOWS\system32	
occache.dll	6.0.3790.1830 Microsoft Corporation	94 KB	3/24/2005 6:08:34 PM	D:\WINDOWS\system32	
proctexe.ocx	6.3.3790.1830 Microsoft Corporation	83 KB	3/24/2005 6:12:26 PM	D:\WINDOWS\system32	Intel Corporation
sendmail.dll	6.0.3790.1830 Microsoft Corporation	56 KB	3/24/2005 6:13:36 PM	D:\WINDOWS\system32	
shdoclc.dll	6.0.3790.0 Microsoft Corporation	589 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32	
shdocvw.dll	6.0.3790.1830 Microsoft Corporation	1,468 KB	3/24/2005 6:13:36 PM	D:\WINDOWS\system32	

shfolder.dll	6.0.3790.1830	25 KB	3/24/2005 6:13:36 PM	D:\WINDOWS\system32
	Microsoft Corporation			
shlwapi.dll	6.0.3790.1830	314 KB	3/24/2005 6:13:40 PM	D:\WINDOWS\system32
	Microsoft Corporation			
tdc.ocx	1.3.0.3130	58 KB	3/25/2003 8:00:00 PM	D:\WINDOWS\system32Microsoft Corporation
url.dll	6.0.3790.1830	37 KB	3/24/2005 6:26:12 PM	D:\WINDOWS\system32Microsoft Corporation
urlmon.dll	6.0.3790.1830	673 KB	3/24/2005 6:26:12 PM	D:\WINDOWS\system32Microsoft Corporation
webcheck.dll	6.0.3790.1830	273 KB	3/24/2005 6:26:16 PM	D:\WINDOWS\system32Microsoft Corporation
wininet.dll	6.0.3790.1830	646 KB	3/24/2005 6:26:18 PM	D:\WINDOWS\system32Microsoft Corporation

[Connectivity]

Item	Value
Connection Preference	Never dial

LAN Settings

AutoConfigProxy	wininet.dll
AutoProxyDetectMode	Disabled
AutoConfigURL	
Proxy	Disabled
ProxyServer	
ProxyOverride	

[Cache]

[Following are sub-categories of this main category]
[Summary]

Item	Value
Page Refresh Type	Automatic
Temporary Internet Files Folder	D:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files
Total Disk Space	Not Available
Available Disk Space	Not Available
Maximum Cache Size	Not Available
Available Cache Size	Not Available

[List of Objects]

Program File	Status	CodeBase
No cached object information available		

[Content]

[Following are sub-categories of this main category]
[Summary]

Item Value
Content Advisor Disabled

[Personal Certificates]

Issued To Issued By Validity Signature Algorithm
No personal certificate information available

[Other People Certificates]

Issued To Issued By Validity Signature Algorithm
No other people certificate information available

[Publishers]

Name
No publisher information available

[Security]

Zone Security Level
My Computer Custom
Local intranet Custom
Trusted sites Custom
Internet Custom
Restricted sites Custom

Microsoft Windows Server 2003 Standard Edition SP1 Client Registry Parameters

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]

"Type"=dword:00000020

"Start"=dword:00000002

"ErrorControl"=dword:00000001

"ImagePath"=hex(2):25,00,53,00,79,00,73,00,74,00,65,00,6d,00,52,00,6f,00,6f,00,\
74,00,25,00,5c,00,53,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,73,\
00,76,00,63,00,68,00,6f,00,73,00,74,00,2e,00,65,00,78,00,65,00,20,00,2d,00,\
6b,00,20,00,69,00,69,00,73,00,73,00,76,00,63,00,73,00,00,00

"DisplayName"="World Wide Web Publishing Service"

"DependOnService"=hex(7):52,00,50,00,43,00,53,00,53,00,00,00,48,00,54,00,54,00,\
50,00,46,00,69,00,6c,00,74,00,65,00,72,00,00,00,49,00,49,00,53,00,41,00,44,\
00,4d,00,49,00,4e,00,00,00,00,00

"DependOnGroup"=hex(7):00,00

"ObjectName"="LocalSystem"

"Description"="Provides Web connectivity and administration through the Internet Information Services Manager"

"FailureActions"=hex:80,51,01,00,01,00,00,00,00,00,00,00,03,00,00,00,53,00,65,\
00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00,01,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]

"MajorVersion"=dword:00000006

```
"MinorVersion"=dword:00000000
"InstallPath"="D:\\WINDOWS\\system32\\inetsrv"
"AccessDeniedMessage"="Error: Access is Denied."
"ServiceDll"=hex(2):44,00,3a,00,5c,00,57,00,49,00,4e,00,44,00,4f,00,57,00,53,\\
00,5c,00,73,00,79,00,73,00,74,00,65,00,6d,00,33,00,32,00,5c,00,69,00,6e,00,\\
65,00,74,00,73,00,72,00,76,00,5c,00,69,00,69,00,73,00,77,00,33,00,61,00,64,\\
00,6d,00,2e,00,64,00,6c,00,6c,00,00,00
"AcceptExOutstanding"=dword:00000028
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunc
h]
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunc
h\\AdvancedDataFactory]
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Parameters\\ADCLaunc
h\\RDSServer.DataFactory]
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Performance]
```

```
"Library"="D:\\WINDOWS\\system32\\inetsrv\\w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"PerfIniFile"="w3ctrs.ini"
"Last Counter"=dword:00000a9e
"Last Help"=dword:00000a9f
"First Counter"=dword:000009a8
"First Help"=dword:000009a9
"Object List"="2472 2646"
"Library Validation Code"=hex:00,5b,bc,13,0a,68,c5,01,00,5e,00,00,00,00,00,00,
"WbemAdapFileSignature"=hex:39,e3,6c,2c,b4,be,59,f5,17,7c,c4,d5,2f,dc,f7,1a
"WbemAdapFileTime"=hex:64,76,15,13,0a,68,c5,01
"WbemAdapFileSize"=dword:00005e00
"WbemAdapStatus"=dword:00000000
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Security]
```

```
"Security"=hex:01,00,14,80,90,00,00,00,9c,00,00,00,14,00,00,00,30,00,00,00,02,\\
00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\\
00,00,02,00,60,00,04,00,00,00,00,00,14,00,fd,01,02,00,01,01,00,00,00,00,00,\\
05,12,00,00,00,00,00,18,00,ff,01,0f,00,01,02,00,00,00,00,00,05,20,00,00,00,\\
20,02,00,00,00,00,14,00,8d,01,02,00,01,01,00,00,00,00,00,05,0b,00,00,00,00,\\
00,18,00,fd,01,02,00,01,02,00,00,00,00,00,05,20,00,00,00,23,02,00,00,01,01,\\
00,00,00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00
```

```
[HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services\\W3SVC\\Enum]
```

```
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Microsoft SQL Server 2005 installation procedure

Microsoft SQL Server 2005 Installation Procedures Type of installation: custom. During the custom

installation, use the default settings for all except the following several areas:

Select the following components:

- SQL Server Database Services

- Notification Services

- Integration Services

- Workstation components, Books Online and development tools

Service Account:

- Use the built-in System account

- Start services at the end of setup: remove ticks for all services

Authentication Mode:

- Mixed mode, and set password to *tpcc*

Collation Settings:

- Collation designator and sort order: Latin1_General Binary

Microsoft COM+ component configuration parameters

The component services tool in Windows Server 2003 Standard 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 the client was 74. Delivery threads were set under the TPCC key in the registry. The construction string was Dummy String.

Benchcraft Profile

Profile: bench

File Path: E:\Documents and Settings\Administrator.BENCH\Desktop\bench.xml

Version: 5

Number of Engines: 8

- Name: mc1-0

- Description: mc1 proc 0

- Directory: c:\tpcc\mc10.log

- Machine: 192.168.0.1

- Parameter Set: testparm

- Index: 0

- Seed: 76264

- Configured Users: 12500

- Pipe Name: DRIVER1227515

- Connect Rate: 200

- Start Rate: 200

- Max. Concurrency: -1

- Concurrency Rate: 10

- CLIENT_NURAND: 233

- CPU: 0

- Additional Options:

- Name: mc1-1

- Description: mc1 proc 1

- Directory: c:\tpcc\mc11.log

- Machine: 192.168.0.1

Parameter Set: testparm
Index: 100000000
Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER2658421
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: mc2-0
Description: mc2 proc 0
Directory: c:\tpcc\mc20.log
Machine: 192.168.0.2
Parameter Set: testparm
Index: 200000000
Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER3708437
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: mc2-1
Description: mc2 proc 1
Directory: c:\tpcc\mc21.log
Machine: 192.168.0.2
Parameter Set: testparm
Index: 300000000
Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER4735046
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: mc3-0
Description:
Directory: c:\tpcc\mc30.log
Machine: 192.168.0.3
Parameter Set: testparm
Index: 400000000

Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER5-1889658843
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: mc3-1
Description:
Directory: c:\tpcc\mc31.log
Machine: 192.168.0.3
Parameter Set: testparm
Index: 500000000
Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER6-1889605218
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Name: mc4-0
Description:
Directory: c:\tpcc\mc40.log
Machine: 192.168.0.4
Parameter Set: testparm
Index: 600000000
Seed: 76264
Configured Users: 12500
Pipe Name: DRIVER7-1889573906
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 0
Additional Options:

Name: mc4-1
Description:
Directory: c:\tpcc\mc41.log
Machine: 192.168.0.4
Parameter Set: testparm
Index: 700000000
Seed: 76264
Configured Users: 12500

Pipe Name: DRIVER8-1889550500
Connect Rate: 200
Start Rate: 200
Max. Concurrency: -1
Concurrency Rate: 10
CLIENT_NURAND: 233
CPU: 1
Additional Options:

Number of User groups: 8

Driver Engine: mc1-0
IIS Server: 192.168.0.171
SQL Server: tpcc1
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1 - 1250
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc1-1
IIS Server: 192.168.0.171
SQL Server: tpcc5
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 1251 - 2500
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc2-0
IIS Server: 192.168.0.172
SQL Server: tpcc2
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 2501 - 3750
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc2-1
IIS Server: 192.168.0.172
SQL Server: tpcc6
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 3751 - 5000
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc3-0
IIS Server: 192.168.0.173
SQL Server: tpcc3
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 5001 - 6250
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc3-1
IIS Server: 192.168.0.173
SQL Server: tpcc7
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 6251 - 7500
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Driver Engine: mc4-0
IIS Server: 192.168.0.174
SQL Server: tpcc4
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 7501 - 8750
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500

District id: 1
Scale Down: No

Driver Engine: mc4-1
IIS Server: 192.168.0.174
SQL Server: tpcc8
Database: tpcc
User: sa
Protocol: HTML
w_id Range: 8751 - 10000
w_id Min Warehouse: 1
w_id Max Warehouse: 10000
Scale: Normal
User Count: 12500
District id: 1
Scale Down: No

Number of Parameter Sets: 3

~Default

Default Parameter Set

	Txn	Think	Key	RT	RT	Menu	
	Weight	Time	Time	Delay	Fence	Delay	
New Order		10.00	12.05	18.01	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

fullspeed

	Txn	Think	Key	RT	RT	Menu	
	Weight	Time	Time	Delay	Fence	Delay	
New Order		44.72	0.00	0.00	0.00	5.00	0.00
Payment		43.08	0.00	0.00	0.00	5.00	0.00
Delivery		4.04	0.00	0.00	0.00	5.00	0.00
Stock Level		4.08	0.00	0.00	0.00	20.00	0.00
Order Status		4.08	0.00	0.00	0.00	5.00	0.00

testparm

	Txn	Think	Key	RT	RT	Menu	
	Weight	Time	Time	Delay	Fence	Delay	
New Order		44.72	12.05	18.02	0.10	5.00	0.10
Payment		43.08	12.05	3.01	0.10	5.00	0.10
Delivery		4.04	5.05	2.01	0.10	5.00	0.10
Stock Level		4.08	5.05	2.01	0.10	20.00	0.10
Order Status		4.08	10.05	2.01	0.10	5.00	0.10

Appendix D: 60-DAY SPACE

TPC-C 60 Day Space Requirements						
Warehouses	10,000	TpmC			125,954	
Table	Rows	Data KB	Index KB	Extra 5% KB	8hr Space	Total Space KB
Warehouse	10,000	1,072	216	64		1352
District	100,000	11,112	320	572		12004
Customer	300,000,000	218,181,840	13,613,568	11,589,770		243385178
History	300,000,000	17,518,256	65,672		3,530,391	17583928
NewOrder	90,000,000	1,603,576	4,168	80,387		1688131
Orders	300,000,000	9,795,936	4,778,840		1,974,140	14574776
OrderLine	2,999,988,197	196,720,560	463,488		39,644,386	197184048
Item	100,000	9,416	312	486		10214
Stock	1,000,000,000	320,000,008	674,848	16,033,743		336708599
Total		763,841,776	19,601,432	27,705,023	45,148,917	811,148,231
		MB				
Dynamic Space	218,784	Sum of Data for Order, Orderline and History				
Static Space	573,353	Sum of Data+Index+5%-Dynamic Space				
Free Space	na	Total Allocated Spac - (Dynamic + Static Space)				
Daily Growth	44,091	(Dynamic Space/(W*62.5))*tpmc				
Daily Spread	-	(Free Space -1.5*Daily Growth) Zero Assumed				
60 Day Space MB	3,218,797					
60 Day Space GB	3,143.36					
Log Size	400,000.00					MB
KB Per New Order	6.75					KB
8 hr log MB	398,459					MB
8 hr log GB	389.1206					GB
		Disks	Disks	Formatted Size	Space	
Space Usage	GB Needed	Measured	Size	Size	Available	
180 Day Space DB	3,143.36	504	36GB	33.890	17080.56	
			36GB		0.00	
			72GB		0.00	
Total DB		504.00			17,080.56	
8-hr log + mirror	778.2411	14	70GB	67.710	947.94	
OS, Swap	3	1	120GB	111.000	111.00	
Total Storage	3,924.60	GB			18,139.50	

The file groups are reported in 8K pages from the sysfile table.

Misc_fg	CS_fg
1352	
12004	
0	243385178
21114319	
1688131	
16548916	
236828434	
10214	
0	336708599
276,203,371	580,093,777
files=	36
size=	1,280,000
Total=	46,080,000
8K blocks	368,640,000
Needed =	276,203,371
OK	OK

tpmC	125,954.00									
	Data Before KB	Index Before KB	Data After KB	Index After KB	Data Grow KB	Index Grow KB	Total Grow KB	KB/New-Order	8-Hr Growth KB	8-Hr Growth MB
History	17,518,256	65,672	19776544	131560	2,258,288	65,888	2,324,176	0.0624	3,772,182.40	3,683.77
Order	9,795,936	4,778,840	12271080	9585648	2,475,144	4,806,808	7,281,952	0.1955	11,818,748.30	11,541.75
Order-Line	196,720,560	463,488	240980216	927416	44,259,656	463,928	44,723,584	1.2006	72,587,237.90	70,885.97
	sum(*) Before	sum(*) After	Num New-Order							
d_next_o_id	300,100,000	337,350,279	37,250,279							
	Before MB	After MB	Grow MB		KB/New-Order	8-Hr Growth MB	8-Hr Growth GB			
Log	3983.28	249488.34	245505.06		6.7489	398,459.44	389.12			
Database tpcc log used (%)					6,910.8400		bytes			
400000	0.99582028	62.372086								

APPENDIX E: Third Party Letters

产品报价 - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 后 前 搜索 收藏夹

地址(D) http://www.skylevel.com.cn/baojia(3com).htm

Google AR2100BLK

Skylevel
天齐科技

天齐首页 网络产品 笔记本产品 服务器产品 软件产品 电源产品 认证培训 系统集成 产品报价 售后服务 关于我们

您的位置: 产品报价/3COM 2006年3月21日星期二

产品报价

- Intel
- Cisco
- 华为3com
 - 3com 交换机
 - 3com 交换机模块
 - 华为交换机
 - 华为交换机模块
 - 华为路由器
 - 华为路由器模块
 - 华为防火墙
 - 华为无线产品
 - 视讯产品
 - APC
 - AVAYA
 - Nortel
 - D-Link

3com 交换机

序号	产品型号	产品描述	用户指导价
1	3C17300	SuperStack 3 Switch 4226T (24 10/100端口+2 10/100/1000端口)	¥ 4,950
2	3C17302	SuperStack 3 Switch 4250T (48 10/100端口+2 10/100/1000端口)	¥ 10,500
3	3C17304	SuperStack 3 Switch 4228G (24 10/100端口+2 10/100/1000端口 + 2 GBIC端口)	¥ 5,600
4	3C17203	SuperStack 3 Switch 4400 24-port (24 10/100 端口)	¥ 10,853
5	3C17204	SuperStack 3 Switch 4400 48-port (48 10/100 端口)	¥ 22,008
6	3C17205	SuperStack 3 Switch 4400 PWR (24 10/100和以太网供电端口)	¥ 16,484
7	3C17206	SuperStack 3 Switch 4400 SE (24 10/100端口)	¥ 5,430
8	3C17210	SuperStack 3 Switch 4400 FX (24 100BASE-FX端口)	¥ 31,274
9	3C17700	SuperStack3 Switch 4900 12-port 100/1000	¥ 27,000
10	3C17701	SuperStack3 Switch 4924 24-port 10/100/1000	¥ 30,500

完毕 Internet

华为 3Com | 服务支持 | 技术服务导航 - Microsoft Internet Explorer

地址(D) http://www.huawei-3com.com.cn/cn/3com/maintainable.htm

产品/产品类型	Small Business	Edge Switching	Personal Systems	Enterprise Core	Software Products
	OfficeConnect 的集线器, 交换机, 网关, 安全路由器, OfficeConnect 无线适配器, PC 卡, Access Points, 防火墙, 路由器, 打印服务器, IntelliJack - FX	SuperStack 交换机: 4900, 4400, 4200, 3800, 3200 系列	IntelliJack (不包括 - FX), Network Jack	交换机: 4005, 40x0, 7700, 8800 系列 路由器: 3000, 5000, 6000 系列 收发器: GBIC, SFP, XFP, XENPAK 安全交换机: 8200, 7200 系列	网络管理, 语音应用软件, 其他应用软件, 操作软件
	Wireless Systems PC 卡	Small Business Baseline 集线器和交换机 (包括 2800, 2200, 2000 系列)	Connectivity 台式机、便携电脑和服务器 NICs 内嵌式防火墙适配器	Voice VCX and MBX 平台, 服务器, 网关, 电话	Wireless Systems 接入点, 路桥, 交换机和控制器 电源适配器, 冗余电源
产品保修期①					
硬件	3 年	终身有限 ③	终身有限 ③	1 年	90 天 (介质更换, 缺陷修复)
维修 / 更换					
产品自带的其它服务 ②					
保修期内硬件维修 / 更换	备件更换 ④	备件更换 ④	备件更换 ④	备件更换 ④	无
电话技术支持	终身有限 ③	终身有限 ③	终身有限 ③	终身有限 ③	终身有限 ③
在线产品	终身有限 ③	终身有限 ③	终身有限 ③	终身有限 ③	终身有限 ③

APC SUA1500RMI2U(电力转换 SUA1500RMI2U) UPS电源 报价 - Microsoft Internet Explorer

文件(E) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) http://detail.zol.com.cn/61/60869/price.shtml

ZOL.com.cn 中关村在线 **新品降价狂拍** **立即加入**

首页 | 新闻中心 | 评测中心 | 行情中心 | 调研中心 | 专题 | 游戏 | 论坛 | 二手 | 报价 | 经销商 | 热评 | 下载 | 搜索 | 电子商城

当前位置: ZOL首页 > 产品频道 > UPS电源 > APC (电力转换) > APC SUA1500RMI2U > 价格

产品一键通

- 笔记本 MP3 手机
- PC电脑 MP4 主板
- DC相机 CPU 显卡
- 摄像机 LCD 机箱
- 摄像头 CRT 硬盘
- 散热器 DVD 内存
- 闪存卡 U盘 音箱
- 打印机 电源 鼠标
- 一体机 键鼠套装
- 扫描仪 数字电视
- 路由器 **更多报价**

APC SUA1500RMI2U

参考价: 3450元

产品类别: UPS电源
生产厂商: APC (电力转换)
类型: 在线互动式
额定输出: 1.5kva
输入电压: 180 - 286V
输入频率: 50/60±3Hz
输出电压: 220 - 240V

暂无推荐评论!

[我来评论](#)

共有0位网友参与评论

中关村在线 ZOL.COM.CN

报价走势图 关注走势图

UPS电源关注排行

- 山特 T6500 ¥175
- 山特 K500 ¥300
- 山特 C1KS ¥1300
- 山特 C3KS ¥2570
- APC BR1000- ¥998
- 山特 C2K ¥3150
- 台达 S 500V ¥230
- 四通 HO-500 ¥180
- 山特 C2KS ¥2400
- 山特 3C10KS ¥13300

品牌关注排行

综述 参数 **报价** 评论 驱动

APC SUA1500RMI2U 报价走势

APC SUA1500RMI2U 价格走势 北京报价

3762

推荐商家

- 利信UPS销售中心
010-82522433 **¥3500**
- 瀚腾电子有限公司
010-82828239 **¥3230**
- 北京益利维斯
010-51855381 **¥3800**
- 北京慧通泰科技
010- 8253573 **¥3550**
- 北京景天奥UPS

SU/MX	
2-3K	公开价(台/年)·使用2年内 1300

备注: 使用未超过两年的UPS可以直接使用以上价格。
使用超过两年的UPS, 需要先通过上门检测, 费用为: RMB600-800/台/次, 再使用以上价格方能进入续保程序

Smart Matrix 系列产品质保服务 (不含电池)

免费维修或换机
1年工作日现场换机服务 (不含法定节假日)
5天*8小时电话技术支持
现场换机响应时间:
24小时内到达现场服务
(北京、上海、广州、成都、南京、深圳、西安、长沙、昆明、福州、乌鲁木齐)

以上价格有效期至2006-12-31止